(12) UK Patent Application (19) GB (11) 2 316 778 (13) A

(43) Date of A Publication 04.03.1998

(21) Application No 9705647.7

(22) Date of Filing 19.03.1997

(30) Priority Data
(31) 08223595 (32) 26.08.1996 (33) JP

(71) Applicant(s)
Fujitsu Limited

(Incorporated in Japan)

1-1 Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211, Japan

(72) Inventor(s)
Takashi Funyu

(74) Agent and/or Address for Service
Haseltine Lake & Co
Imperial House, 15-19 Kingsway, LONDON, WC2B 6UD, United Kingdom

(51) INT CL[6]
G06F 17/21 15/163 17/30

(52) UK CL (Edition P)
G4A AFGDC
H4T TDAA

(56) Documents Cited
GB 2051527 A GB 2022378 A EP 0511438 A1
MacWEEK, Vol 10 No 15, 15/4/1996, Rebecca Gulick, "Vendors vie for Web font standard", pages 12-13

(58) Field of Search
UK CL (Edition O) B6F FAM, G4A AFGDC, H4T TDAA
INT CL[6] G06F 15/163 17/21 17/30
Online: WPI, COMPUTER, BUSSOFT

(54) World Wide Web server sending only required subset of font

(57) When character media data is output to a client unit, a server unit dynamically extracts font data corresponding to the character media data from font resources of various styles of type that have been prepared in advance and creates required font resources. The font resources are sent to the client unit through the network and are used to display the character media data.

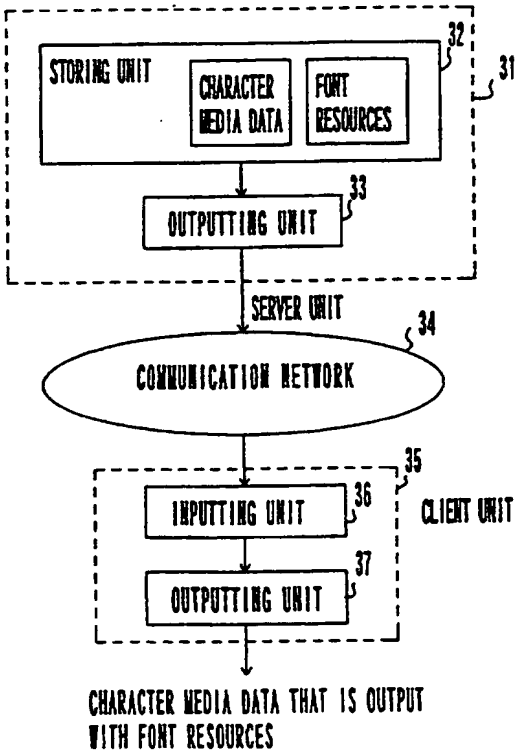

FIG. 2A

GB 2 316 778 A

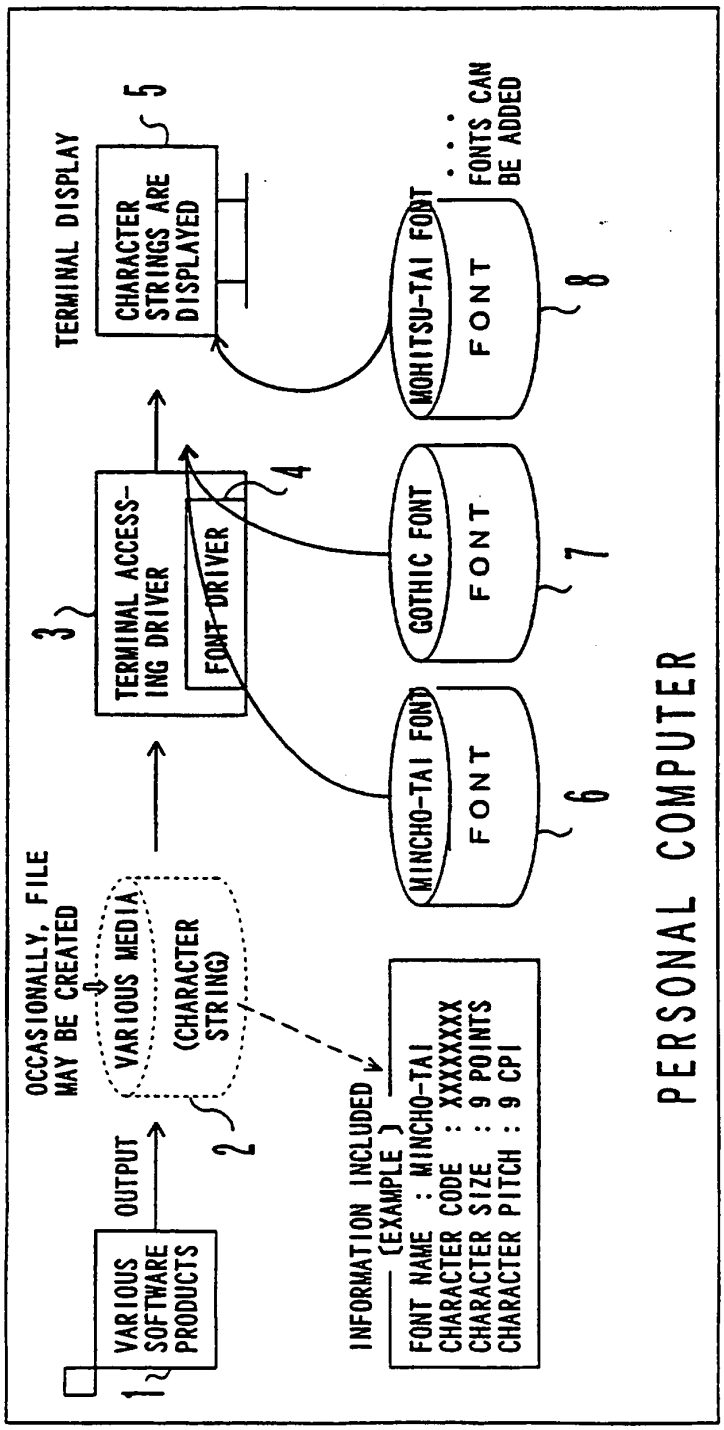

FIG. 1A

ISDOCID: <GB___2316778A_I_>

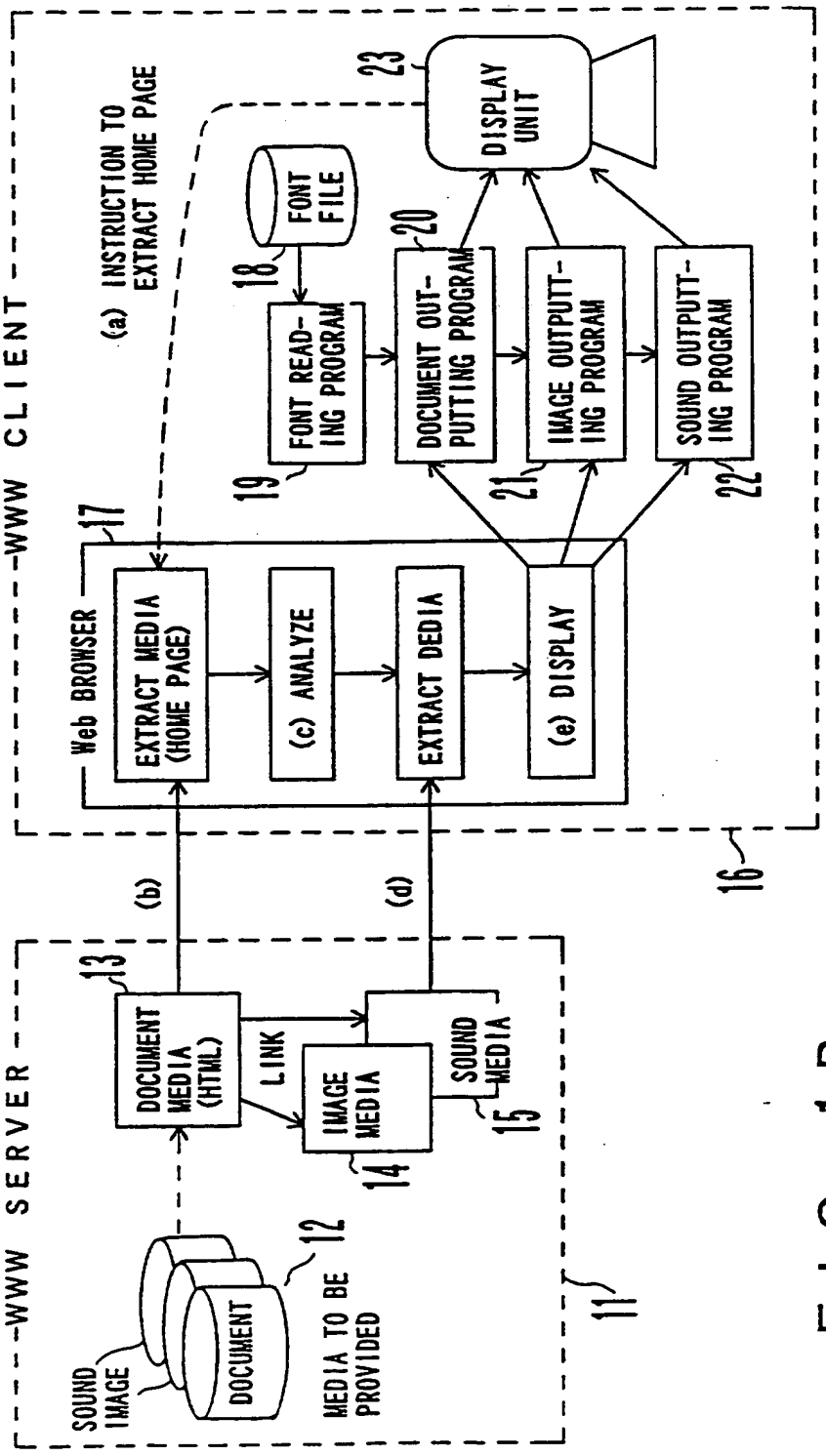
-WWW SERVER-
SOUND
IMAGE
DOCUMENT
MEDIA TO BE PROVIDED
12
11
DOCUMENT MEDIA (HTML)
13
LINK
IMAGE MEDIA
14
SOUND MEDIA
15
(b)
(d)
-WWW CLIENT-
16
Web BROWSER
17
EXTRACT MEDIA (HOME PAGE)
(c) ANALYZE
EXTRACT DEDIA
(e) DISPLAY
(a) INSTRUCTION TO EXTRACT HOME PAGE
FONT FILE
18
FONT READ-ING PROGRAM
19
DOCUMENT OUT-PUTTING PROGRAM
20
IMAGE OUTPUTT-ING PROGRAM
21
SOUND OUTPUTT-ING PROGRAM
22
DISPLAY UNIT
23

FIG. 1B

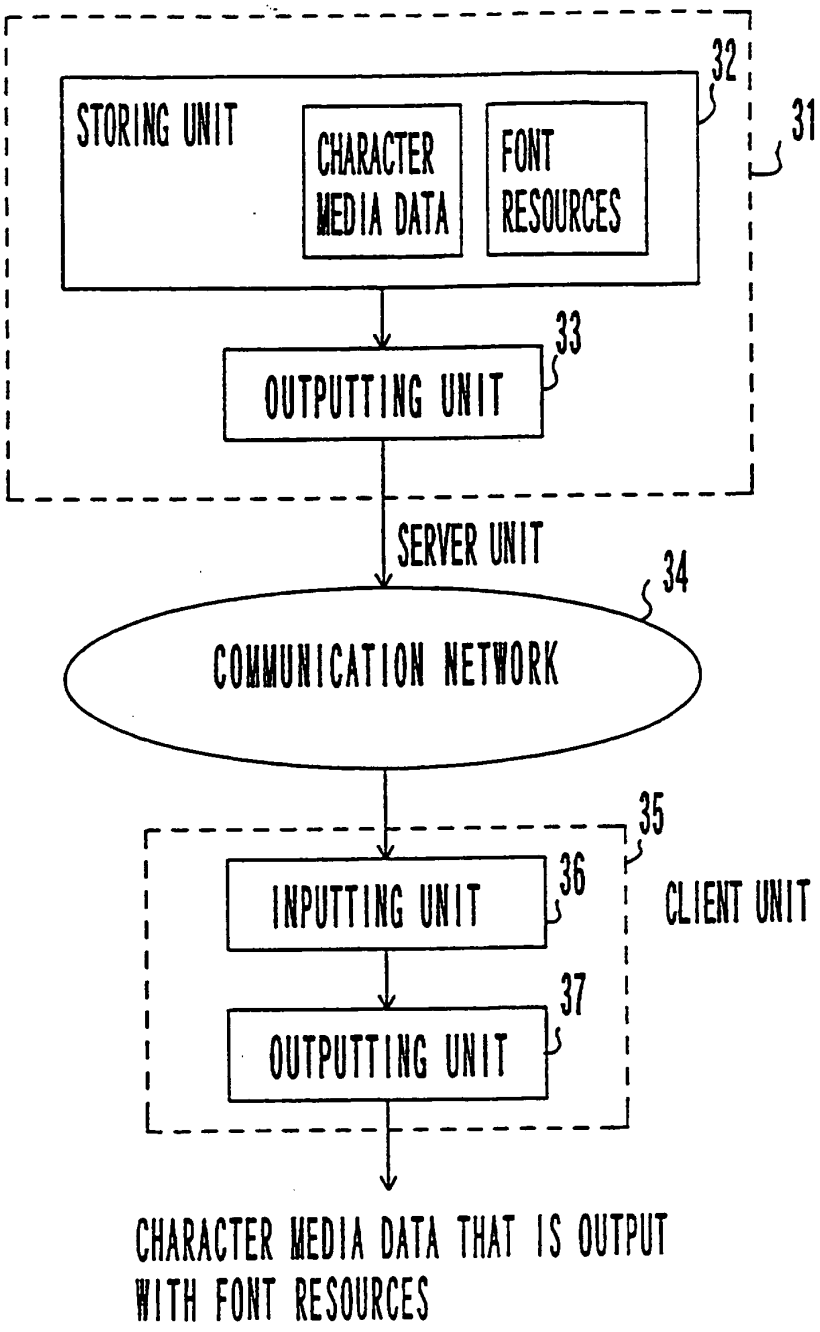

FIG. 2A

PNSDOCID: <GB 2316778A 1 >

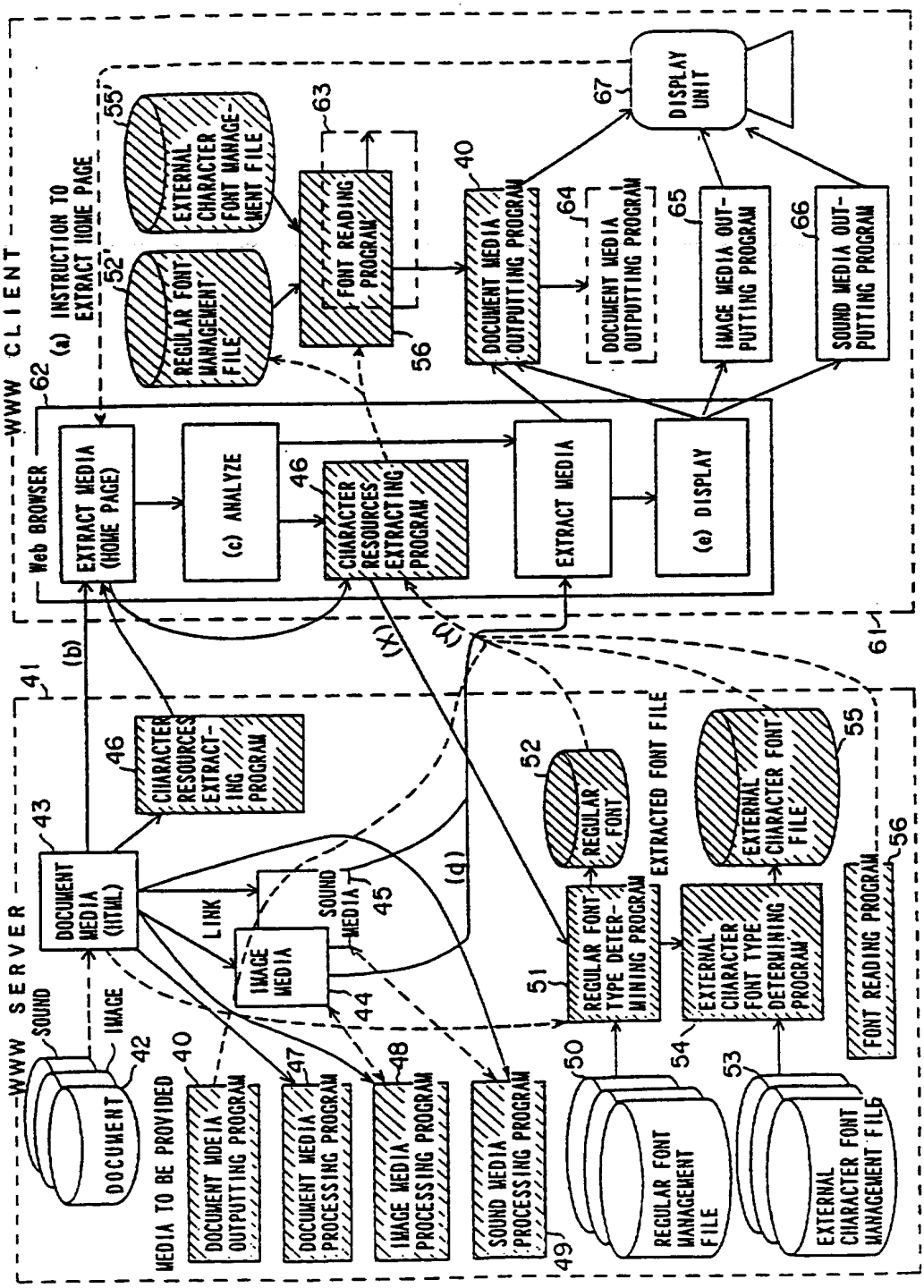
WWW SERVER
WWW CLIENT
Web BROWSER
(a) INSTRUCTION TO EXTRACT HOME PAGE
EXTRACT MEDIA (HOME PAGE)
(c) ANALYZE
CHARACTER RESOURCES EXTRACTING PROGRAM
EXTRACT MEDIA
(e) DISPLAY
DISPLAY UNIT
EXTERNAL CHARACTER FONT MANAGEMENT FILE
REGULAR FONT MANAGEMENT FILE
FONT READING PROGRAM
DOCUMENT MEDIA OUTPUTTING PROGRAM
IMAGE MEDIA OUTPUTTING PROGRAM
SOUND MEDIA OUTPUTTING PROGRAM
DOCUMENT MEDIA (HTML)
LINK
IMAGE MEDIA
SOUND MEDIA
REGULAR FONT
EXTRACTED FONT FILE
EXTERNAL CHARACTER FONT FILE
REGULAR FONT TYPE DETERMINING PROGRAM
EXTERNAL CHARACTER FONT TYPE DETERMINING PROGRAM
FONT READING PROGRAM
MEDIA TO BE PROVIDED
DOCUMENT
SOUND
IMAGE
DOCUMENT MEDIA OUTPUTTING PROGRAM
DOCUMENT MEDIA PROCESSING PROGRAM
IMAGE MEDIA PROCESSING PROGRAM
SOUND MEDIA PROCESSING PROGRAM
REGULAR FONT MANAGEMENT FILE
EXTERNAL CHARACTER FONT MANAGEMENT FILE
(b)
(d)
(X)
(Y)

FIG. 2B

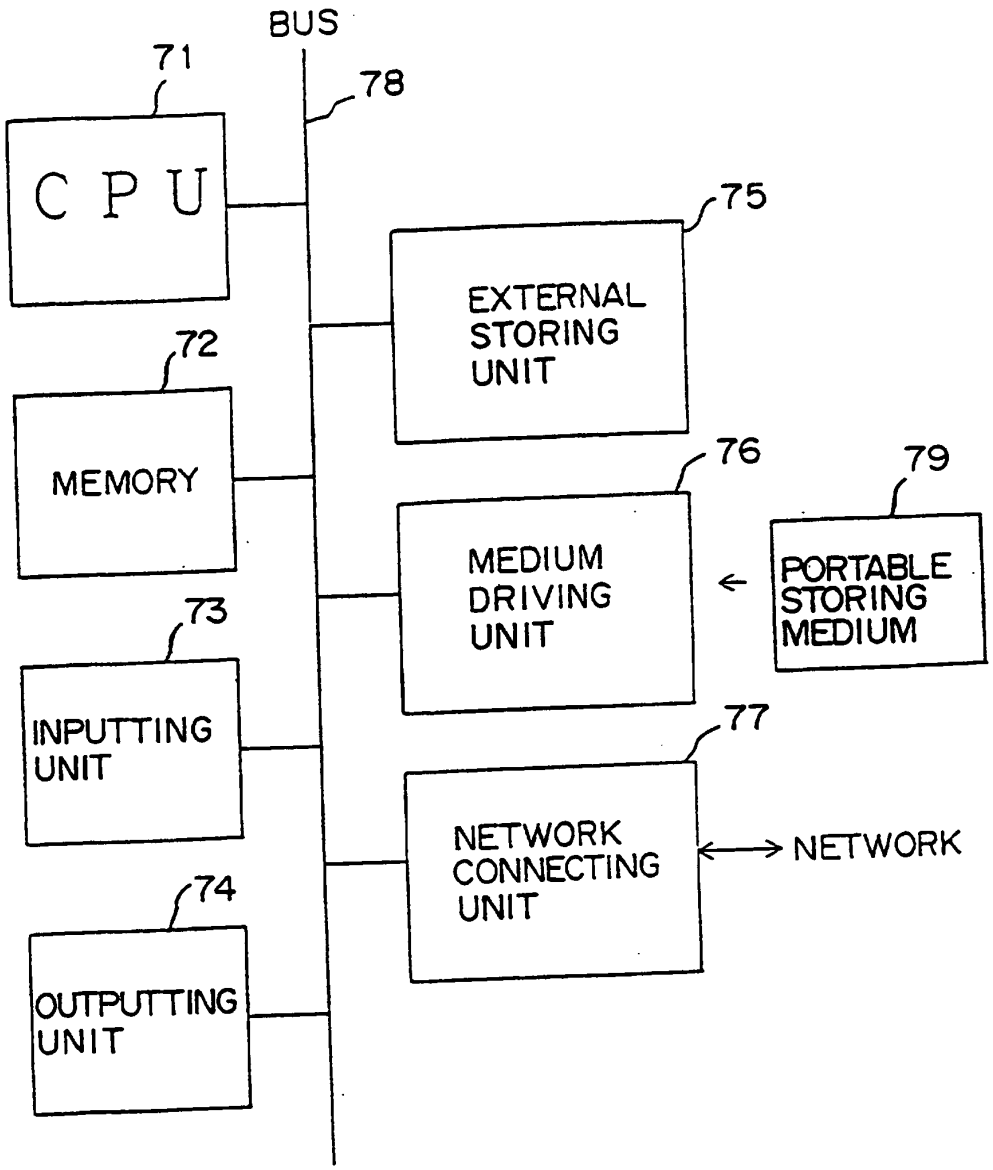
BUS
71
78
C P U
75
EXTERNAL STORING UNIT
72
MEMORY
76
79
MEDIUM DRIVING UNIT
PORTABLE STORING MEDIUM
73
INPUTTING UNIT
77
NETWORK CONNECTING UNIT
NETWORK
74
OUTPUTTING UNIT

FIG. 3

```
----------HOME PAGE OF VARIOUS SERVICES (HTML DOCUMENT)----------

  <HTML>
  <HEAD>
  <TITLE>  HOME PAGE (SAMPLE) </TITLE>
  </HEAD>
  <BODY>
    THIS COMPANY WILL PROVIDE THE FOLLOWING SERVICES. <BR>
                          .
                          .
                          .
    L1~ <APPLET CODE='FontGet.class'> ···········→ DESIGNATE CHARACTER RESOURCES
                                                   EXTRACTING APPLET.
    L2~ <PARAM NAME=text value='http://www.fujitsu.co.jp/Service.html'>  ····→ URL OF
                                                                          HTML DOCUMENT
        </APPLET>

  <H1> OUTLINE OF SERVICE</H1>
                    .
                    .
                    .
L3~ <APPLET CODEBASE='http://www.fujitsu.co.jp/'      ··········→LOCATION OF APPLET
L4~         CODE='TextDraw.class' WIDTH=200 HEIGHT=50>  ··········→DESIGNATE DOCUMENT
                                                          MEDIA PROCESSING APPLET
L5~ <PARAM NAME='text' value='CHARACTER STRING'>       ··········→DESIGNATE
                                                          CHARACTER MEDIA
  </APPLET>

L6~ <APPLET CODE='ImageDraw. class'>                   ··········→DESIGNATE IMAGE
                                                          MEDIA PROCESSING APPLET
L7~ <PARAM NAME='image' IMG='Image.gif'>               ··········→DESIGNATE IMAGE
                                                          MEDIA
  </APPLET>

L8~ <APPLET CODE='VoiceDraw. class'>                   ··········→DESIGNATE SOUND
                                                          MEDIA PROCESSING APPLET
L9~ <PARAM NAME='midi' MIDI='Voice.aiff'>              ··········→DESIGNATE SOUND
                                                          MEDIA
  </APPLET>
                    .
                    .
                    .
  </BODY>
  </HTML>
```

FIG. 4

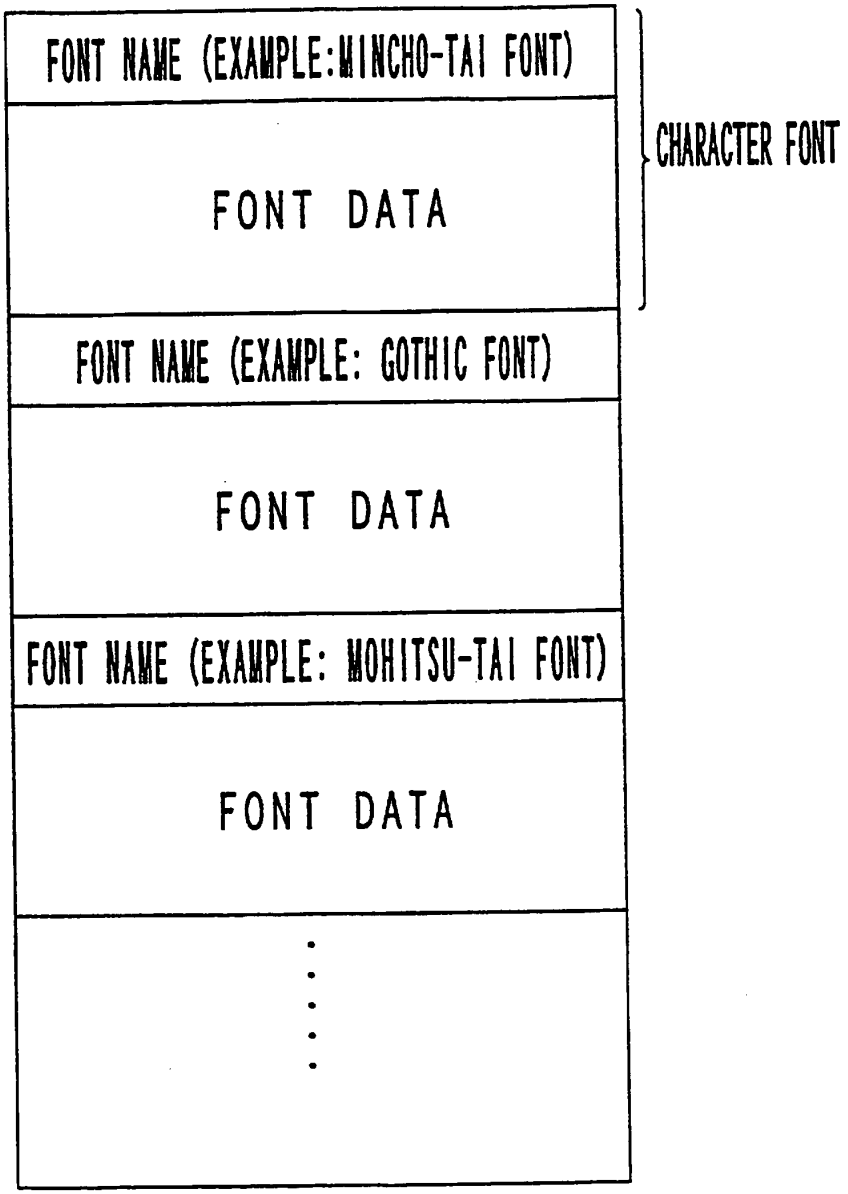
FONT NAME (EXAMPLE:MINCHO-TAI FONT)
FONT DATA
CHARACTER FONT
FONT NAME (EXAMPLE: GOTHIC FONT)
FONT DATA
FONT NAME (EXAMPLE: MOHITSU-TAI FONT)
FONT DATA

FIG. 5

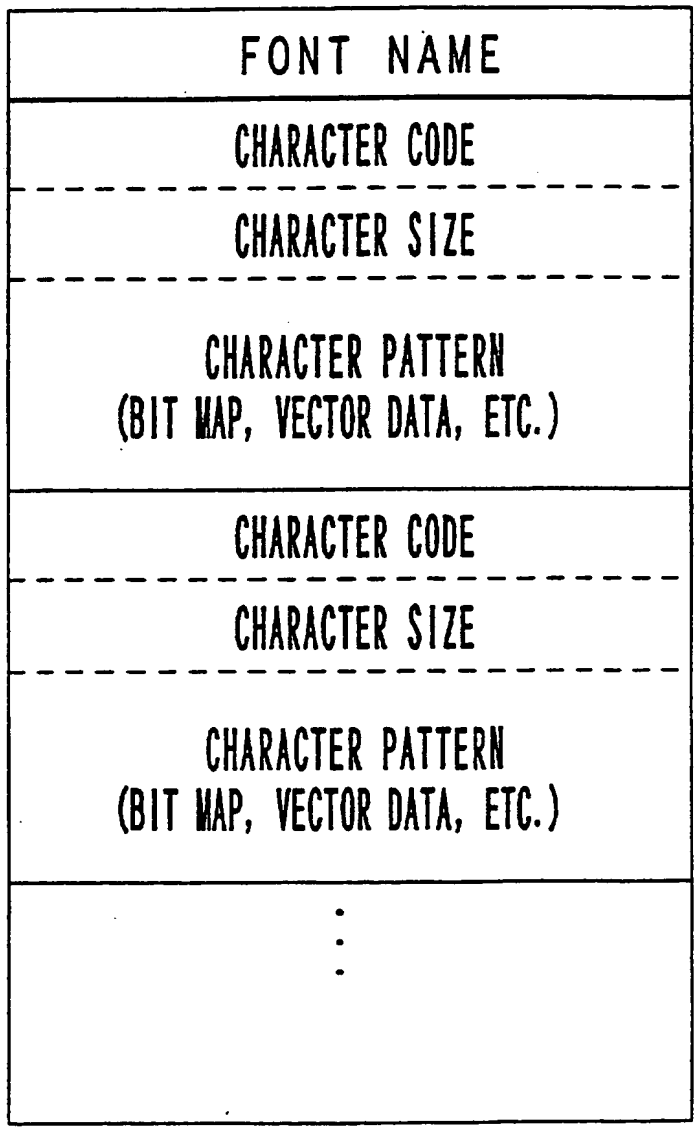
FONT NAME
CHARACTER CODE
CHARACTER SIZE
CHARACTER PATTERN
(BIT MAP, VECTOR DATA, ETC.)
CHARACTER CODE
CHARACTER SIZE
CHARACTER PATTERN
(BIT MAP, VECTOR DATA, ETC.)

FIG. 6

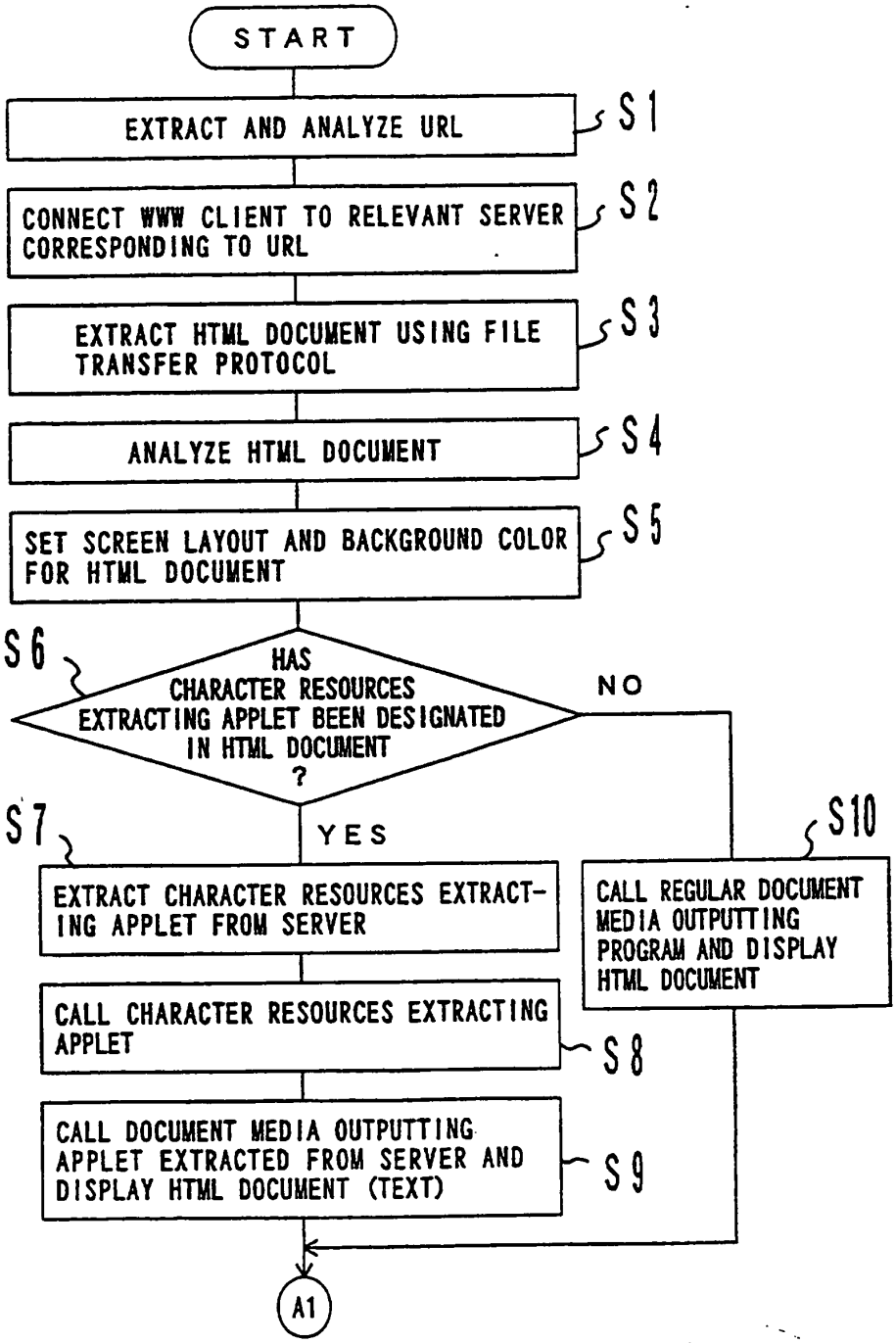
START
EXTRACT AND ANALYZE URL
S1
CONNECT WWW CLIENT TO RELEVANT SERVER CORRESPONDING TO URL
S2
EXTRACT HTML DOCUMENT USING FILE TRANSFER PROTOCOL
S3
ANALYZE HTML DOCUMENT
S4
SET SCREEN LAYOUT AND BACKGROUND COLOR FOR HTML DOCUMENT
S5
S6
HAS CHARACTER RESOURCES EXTRACTING APPLET BEEN DESIGNATED IN HTML DOCUMENT ?
NO
YES
S7
EXTRACT CHARACTER RESOURCES EXTRACTING APPLET FROM SERVER
CALL CHARACTER RESOURCES EXTRACTING APPLET
S8
CALL DOCUMENT MEDIA OUTPUTTING APPLET EXTRACTED FROM SERVER AND DISPLAY HTML DOCUMENT (TEXT)
S9
S10
CALL REGULAR DOCUMENT MEDIA OUTPUTTING PROGRAM AND DISPLAY HTML DOCUMENT
A1

FIG. 7

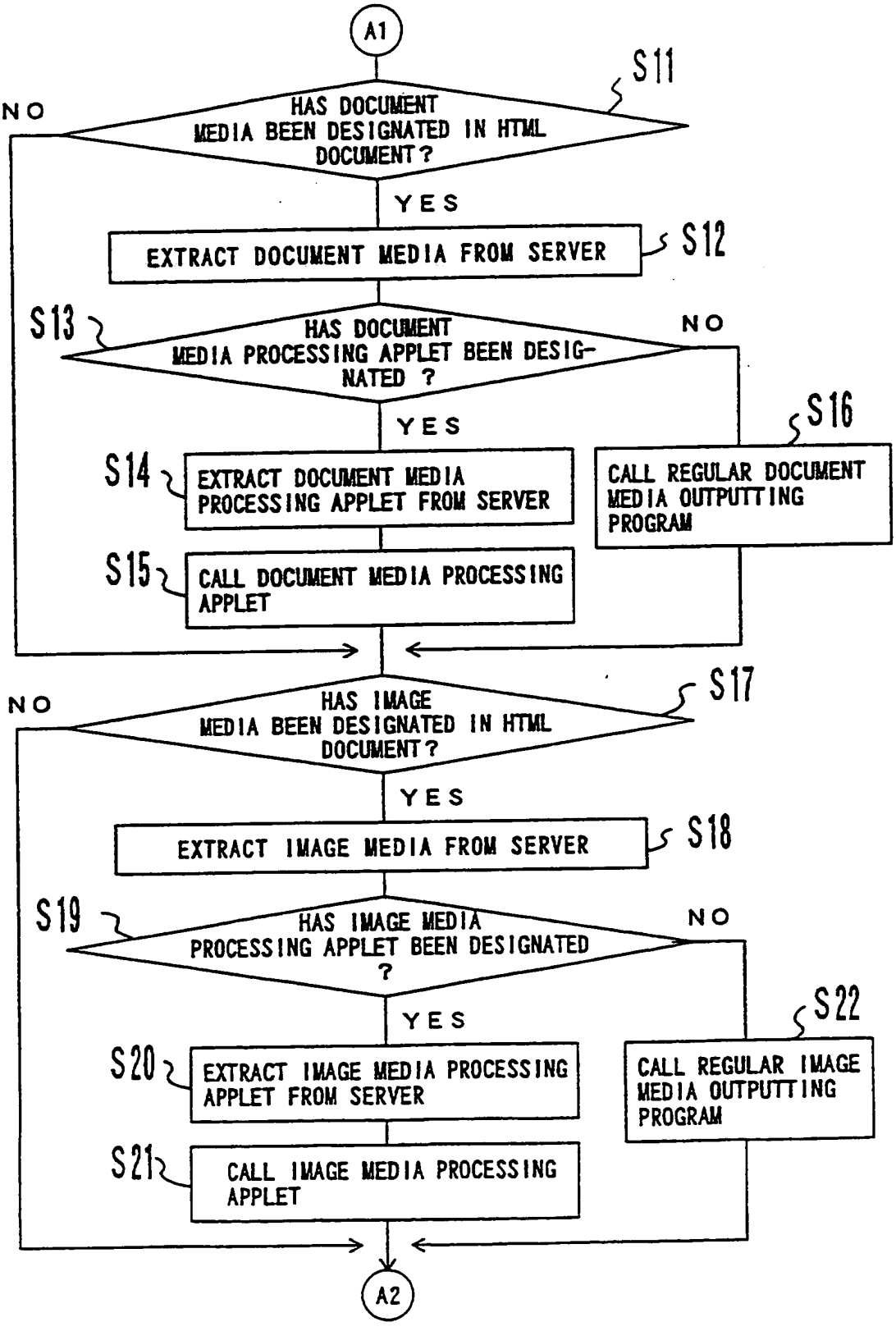
A1
S11
HAS DOCUMENT MEDIA BEEN DESIGNATED IN HTML DOCUMENT?
NO
YES
EXTRACT DOCUMENT MEDIA FROM SERVER
S12
S13
HAS DOCUMENT MEDIA PROCESSING APPLET BEEN DESIGNATED ?
NO
YES
S16
CALL REGULAR DOCUMENT MEDIA OUTPUTTING PROGRAM
S14
EXTRACT DOCUMENT MEDIA PROCESSING APPLET FROM SERVER
S15
CALL DOCUMENT MEDIA PROCESSING APPLET
S17
HAS IMAGE MEDIA BEEN DESIGNATED IN HTML DOCUMENT?
NO
YES
EXTRACT IMAGE MEDIA FROM SERVER
S18
S19
HAS IMAGE MEDIA PROCESSING APPLET BEEN DESIGNATED ?
NO
YES
S22
CALL REGULAR IMAGE MEDIA OUTPUTTING PROGRAM
S20
EXTRACT IMAGE MEDIA PROCESSING APPLET FROM SERVER
S21
CALL IMAGE MEDIA PROCESSING APPLET
A2

FIG. 8

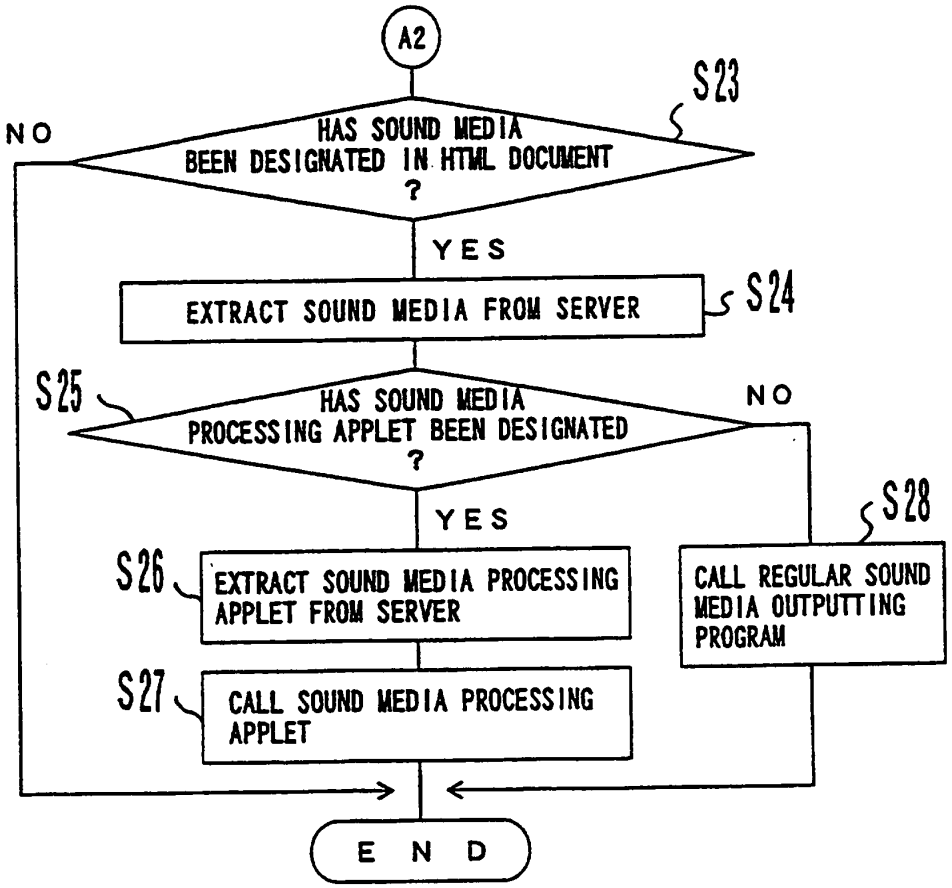

FIG. 9

BNSDOCID: <GB 2316778A I >

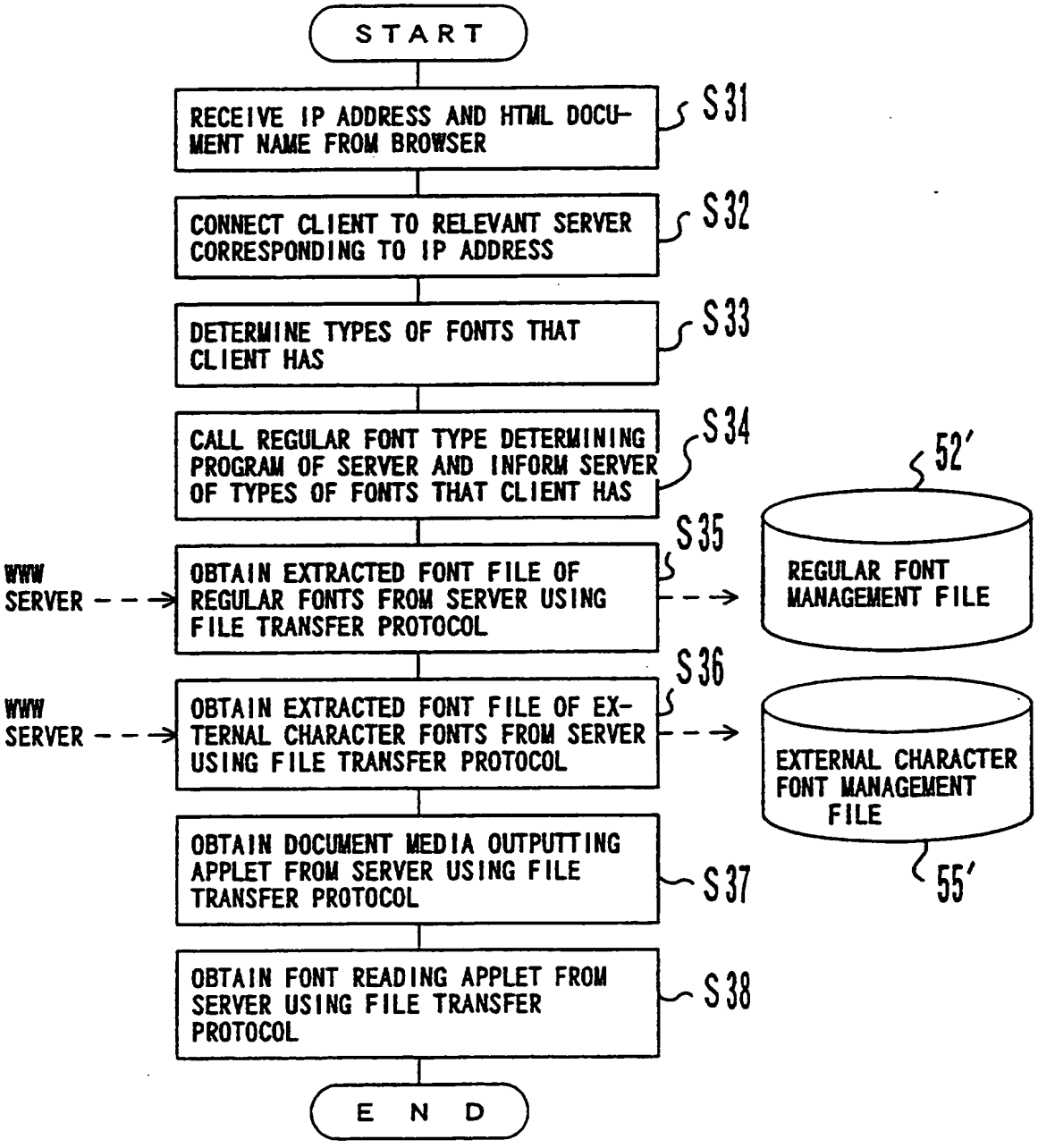
START
RECEIVE IP ADDRESS AND HTML DOCUMENT NAME FROM BROWSER
S31
CONNECT CLIENT TO RELEVANT SERVER CORRESPONDING TO IP ADDRESS
S32
DETERMINE TYPES OF FONTS THAT CLIENT HAS
S33
CALL REGULAR FONT TYPE DETERMINING PROGRAM OF SERVER AND INFORM SERVER OF TYPES OF FONTS THAT CLIENT HAS
S34
52'
WWW SERVER
OBTAIN EXTRACTED FONT FILE OF REGULAR FONTS FROM SERVER USING FILE TRANSFER PROTOCOL
S35
REGULAR FONT MANAGEMENT FILE
WWW SERVER
OBTAIN EXTRACTED FONT FILE OF EXTERNAL CHARACTER FONTS FROM SERVER USING FILE TRANSFER PROTOCOL
S36
EXTERNAL CHARACTER FONT MANAGEMENT FILE
55'
OBTAIN DOCUMENT MEDIA OUTPUTTING APPLET FROM SERVER USING FILE TRANSFER PROTOCOL
S37
OBTAIN FONT READING APPLET FROM SERVER USING FILE TRANSFER PROTOCOL
S38
END

FIG. 10

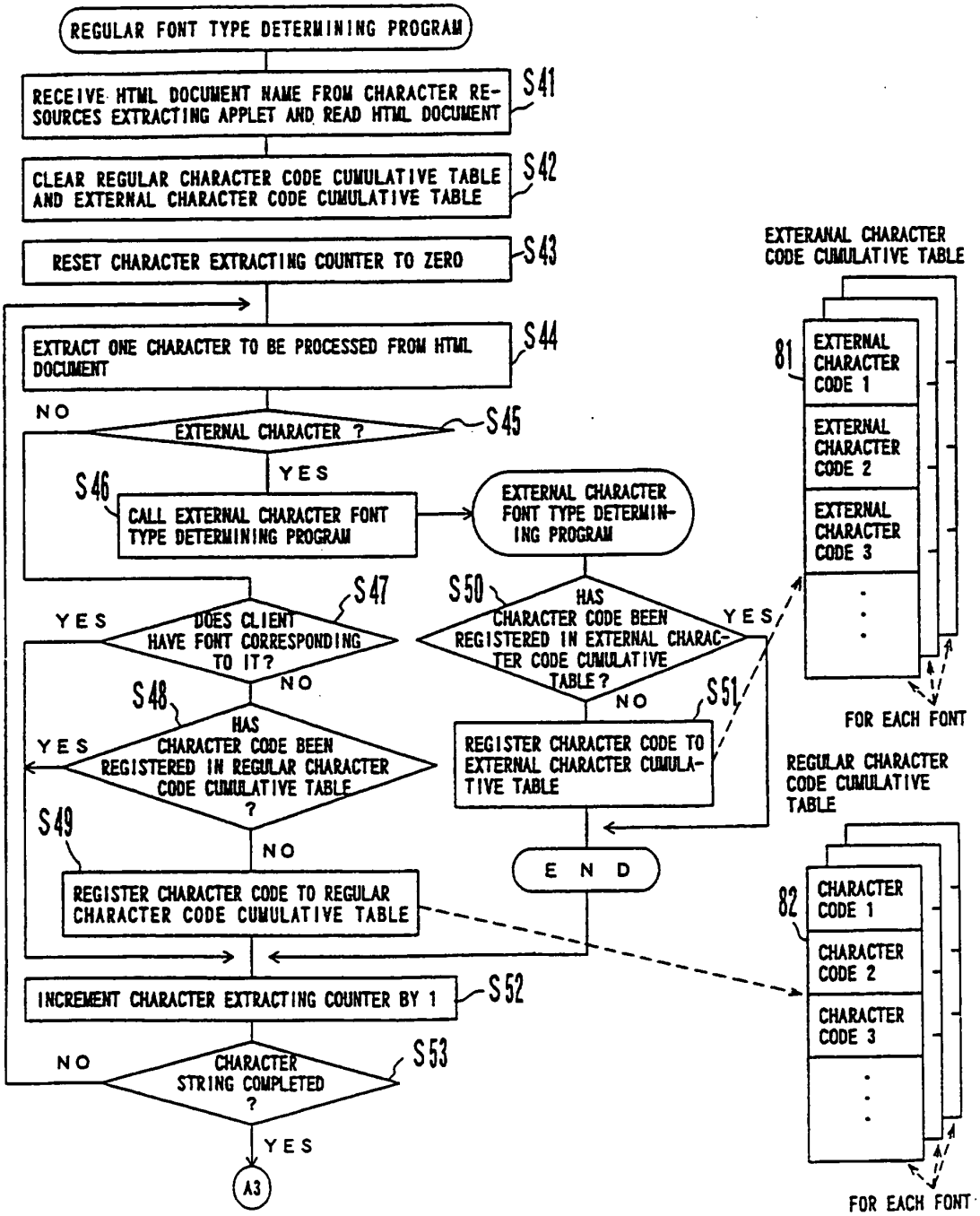
REGULAR FONT TYPE DETERMINING PROGRAM
RECEIVE HTML DOCUMENT NAME FROM CHARACTER RESOURCES EXTRACTING APPLET AND READ HTML DOCUMENT
S41
CLEAR REGULAR CHARACTER CODE CUMULATIVE TABLE AND EXTERNAL CHARACTER CODE CUMULATIVE TABLE
S42
RESET CHARACTER EXTRACTING COUNTER TO ZERO
S43
EXTRACT ONE CHARACTER TO BE PROCESSED FROM HTML DOCUMENT
S44
NO
EXTERNAL CHARACTER ?
S45
YES
S46
CALL EXTERNAL CHARACTER FONT TYPE DETERMINING PROGRAM
EXTERNAL CHARACTER FONT TYPE DETERMINING PROGRAM
S47
YES
DOES CLIENT HAVE FONT CORRESPONDING TO IT?
NO
S48
HAS CHARACTER CODE BEEN REGISTERED IN REGULAR CHARACTER CODE CUMULATIVE TABLE ?
YES
S49
NO
REGISTER CHARACTER CODE TO REGULAR CHARACTER CODE CUMULATIVE TABLE
S50
HAS CHARACTER CODE BEEN REGISTERED IN EXTERNAL CHARACTER CODE CUMULATIVE TABLE?
YES
NO
S51
REGISTER CHARACTER CODE TO EXTERNAL CHARACTER CUMULATIVE TABLE
E N D
INCREMENT CHARACTER EXTRACTING COUNTER BY 1
S52
NO
CHARACTER STRING COMPLETED ?
S53
YES
A3
EXTERANAL CHARACTER CODE CUMULATIVE TABLE
81
EXTERNAL CHARACTER CODE 1
EXTERNAL CHARACTER CODE 2
EXTERNAL CHARACTER CODE 3
FOR EACH FONT
REGULAR CHARACTER CODE CUMULATIVE TABLE
82
CHARACTER CODE 1
CHARACTER CODE 2
CHARACTER CODE 3
FOR EACH FONT

FIG. 11

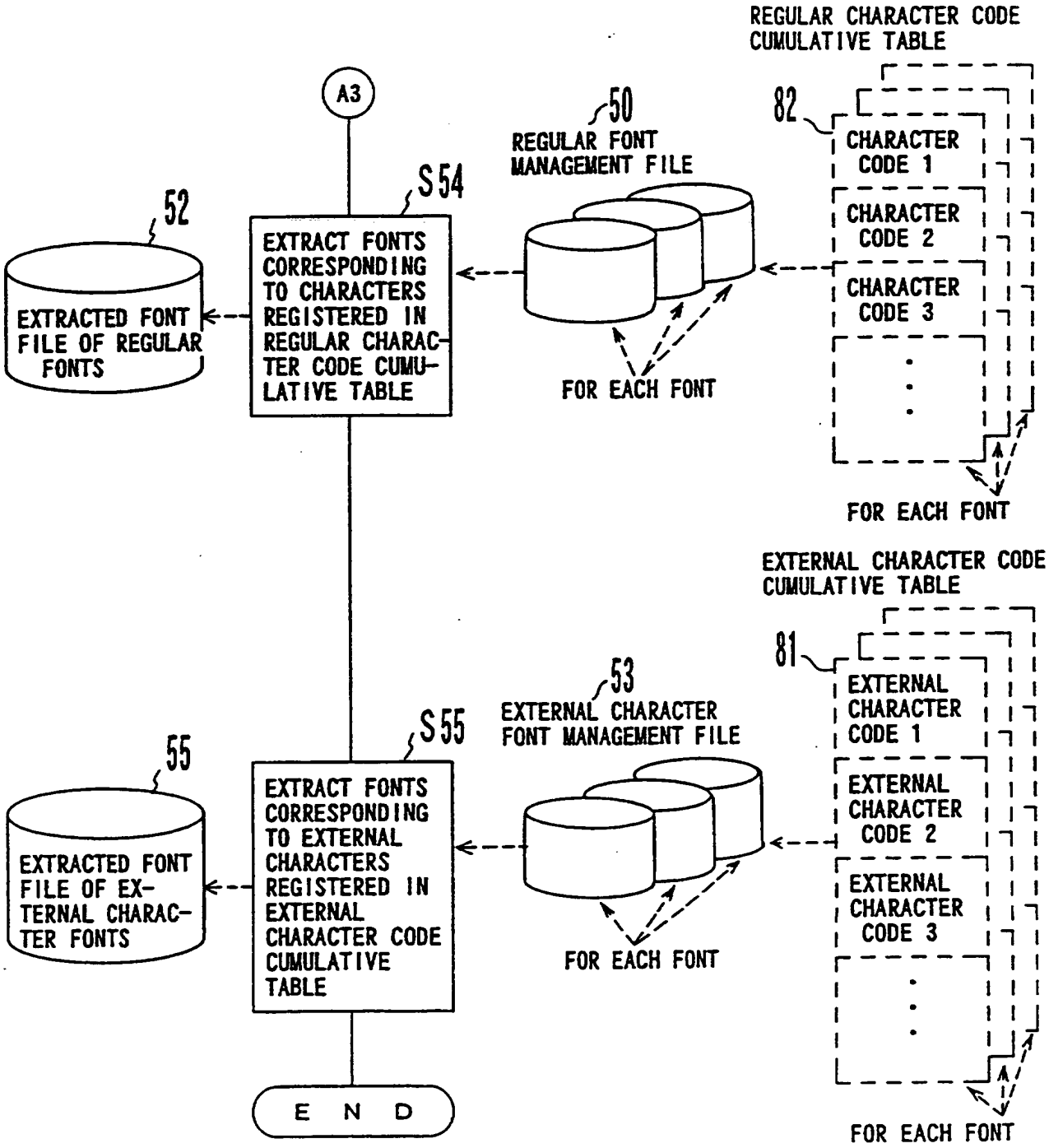
A3
S54
EXTRACT FONTS CORRESPONDING TO CHARACTERS REGISTERED IN REGULAR CHARACTER CODE CUMULATIVE TABLE
52
EXTRACTED FONT FILE OF REGULAR FONTS
50
REGULAR FONT MANAGEMENT FILE
FOR EACH FONT
REGULAR CHARACTER CODE CUMULATIVE TABLE
82
CHARACTER CODE 1
CHARACTER CODE 2
CHARACTER CODE 3
FOR EACH FONT
S55
EXTRACT FONTS CORRESPONDING TO EXTERNAL CHARACTERS REGISTERED IN EXTERNAL CHARACTER CODE CUMULATIVE TABLE
55
EXTRACTED FONT FILE OF EXTERNAL CHARACTER FONTS
53
EXTERNAL CHARACTER FONT MANAGEMENT FILE
FOR EACH FONT
EXTERNAL CHARACTER CODE CUMULATIVE TABLE
81
EXTERNAL CHARACTER CODE 1
EXTERNAL CHARACTER CODE 2
EXTERNAL CHARACTER CODE 3
FOR EACH FONT
E N D

FIG. 12

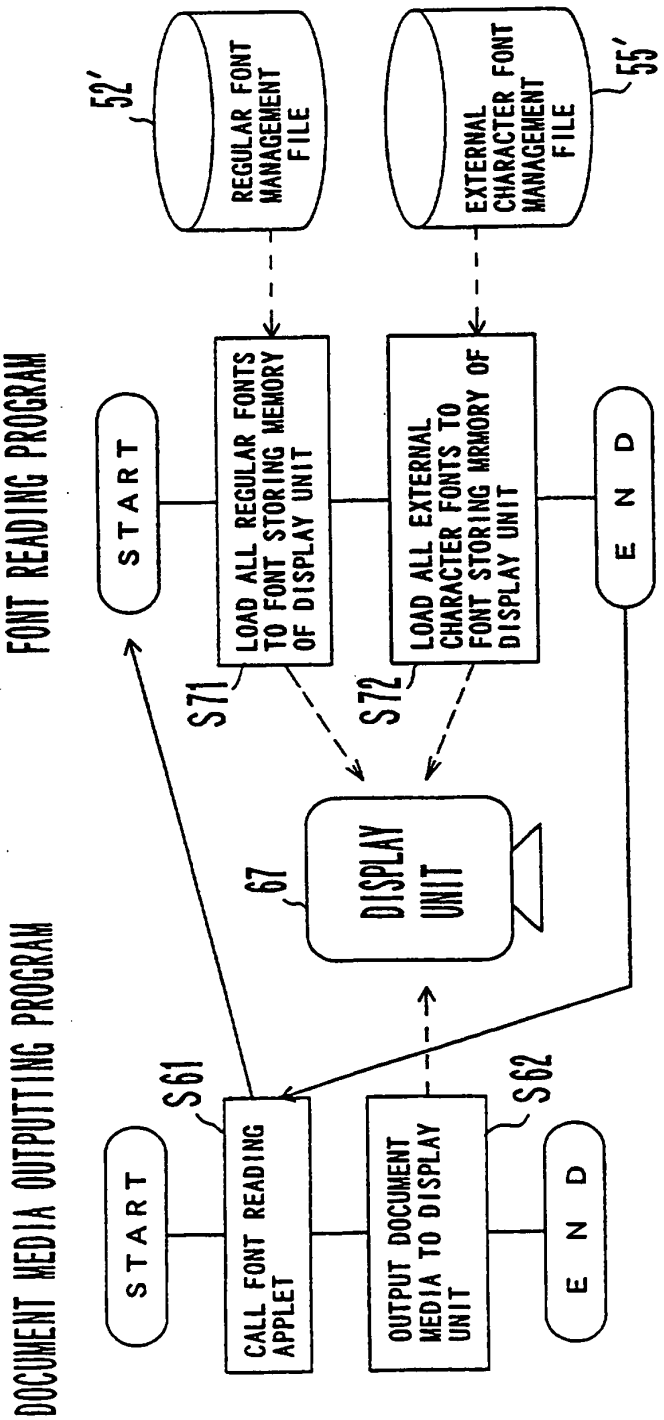
DOCUMENT MEDIA OUTPUTTING PROGRAM
FONT READING PROGRAM
START
CALL FONT READING APPLET
S61
OUTPUT DOCUMENT MEDIA TO DISPLAY UNIT
S62
END
DISPLAY UNIT
67
START
LOAD ALL REGULAR FONTS TO FONT STORING MEMORY OF DISPLAY UNIT
S71
LOAD ALL EXTERNAL CHARACTER FONTS TO FONT STORING MRMORY OF DISPLAY UNIT
S72
END
52'
REGULAR FONT MANAGEMENT FILE
55'
EXTERNAL CHARACTER FONT MANAGEMENT FILE

FIG. 13

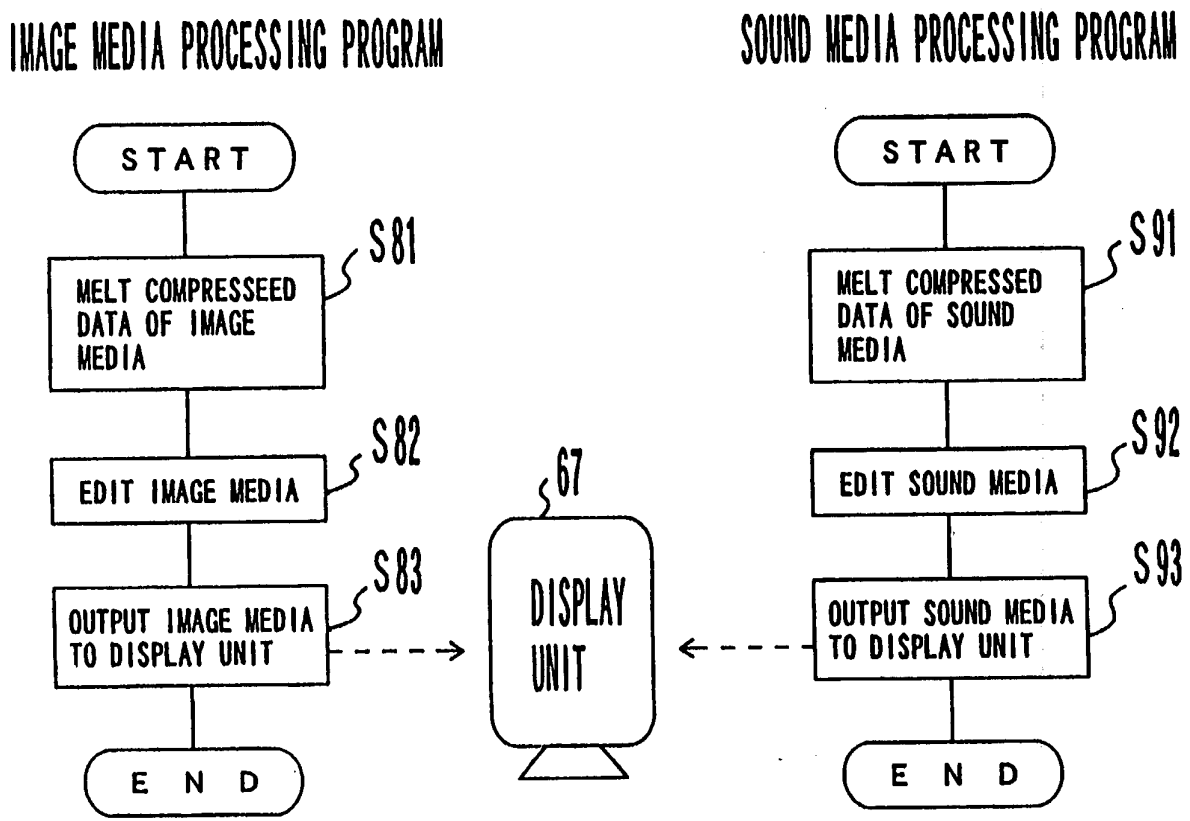
IMAGE MEDIA PROCESSING PROGRAM
START
S81
MELT COMPRESSEED DATA OF IMAGE MEDIA
S82
EDIT IMAGE MEDIA
S83
OUTPUT IMAGE MEDIA TO DISPLAY UNIT
E N D
67
DISPLAY UNIT
SOUND MEDIA PROCESSING PROGRAM
START
S91
MELT COMPRESSED DATA OF SOUND MEDIA
S92
EDIT SOUND MEDIA
S93
OUTPUT SOUND MEDIA TO DISPLAY UNIT
E N D

F I G. 1 4

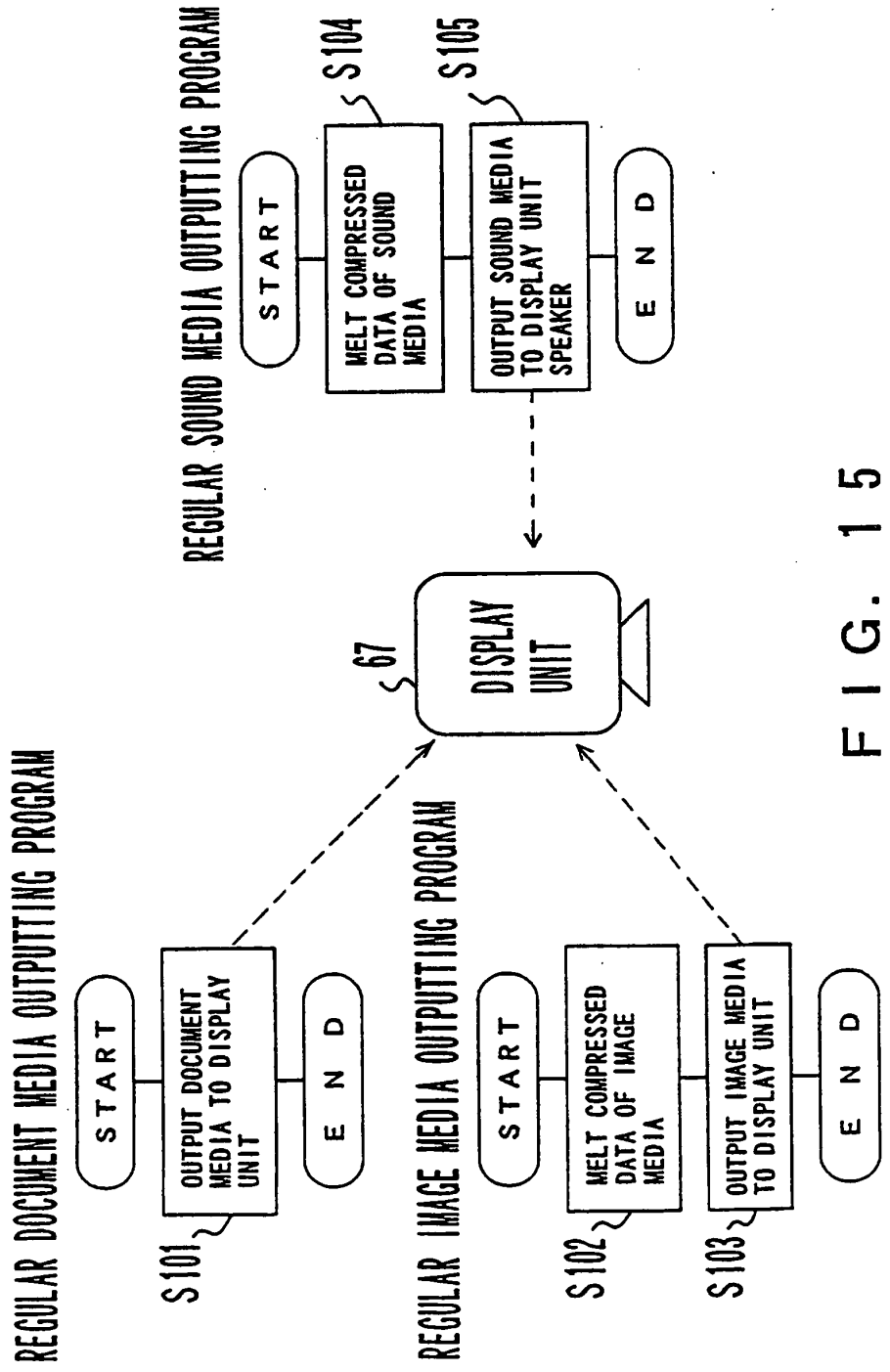
REGULAR DOCUMENT MEDIA OUTPUTTING PROGRAM
START
OUTPUT DOCUMENT MEDIA TO DISPLAY UNIT
END
S101
REGULAR IMAGE MEDIA OUTPUTTING PROGRAM
START
MELT COMPRESSED DATA OF IMAGE MEDIA
OUTPUT IMAGE MEDIA TO DISPLAY UNIT
END
S102
S103
67
DISPLAY UNIT
REGULAR SOUND MEDIA OUTPUTTING PROGRAM
START
MELT COMPRESSED DATA OF SOUND MEDIA
OUTPUT SOUND MEDIA TO DISPLAY UNIT SPEAKER
END
S104
S105

FIG. 15

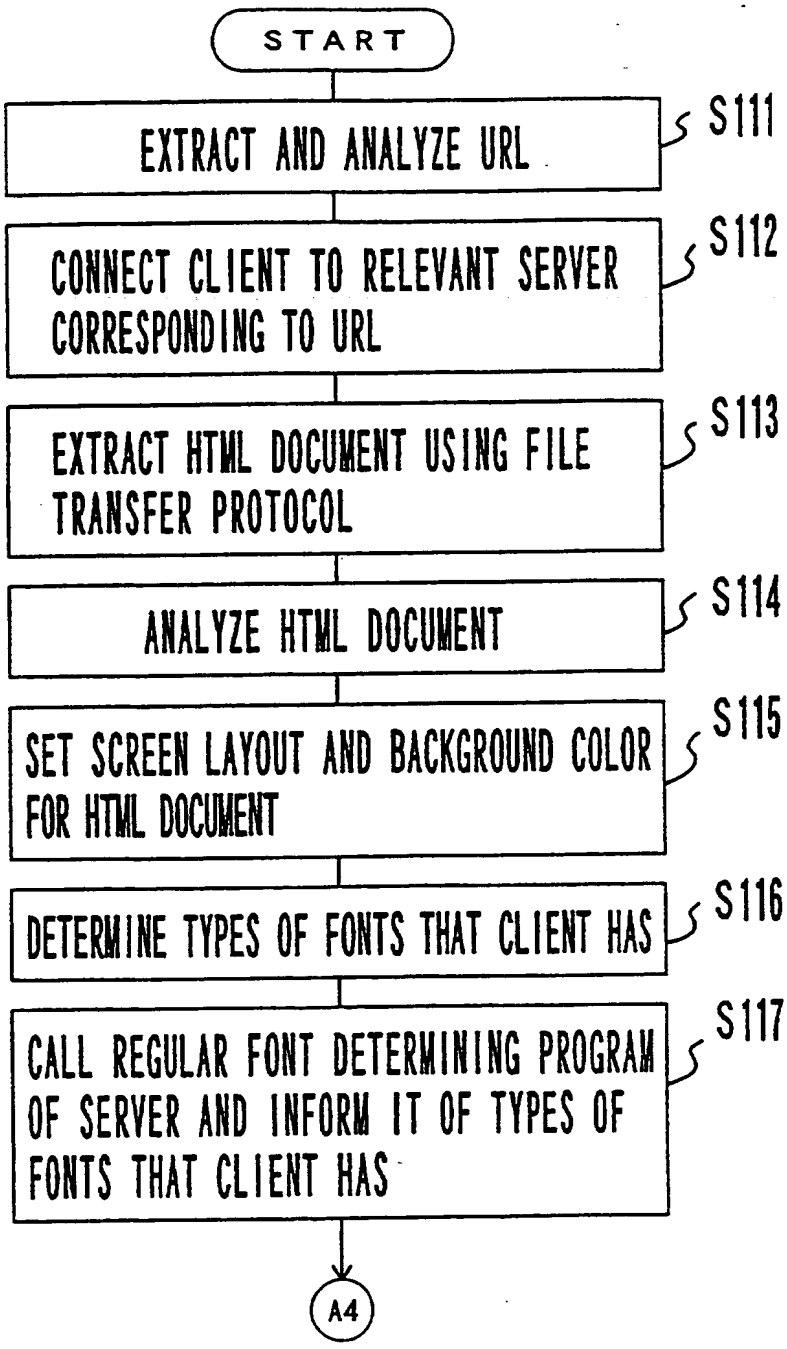
START
EXTRACT AND ANALYZE URL
S111
CONNECT CLIENT TO RELEVANT SERVER CORRESPONDING TO URL
S112
EXTRACT HTML DOCUMENT USING FILE TRANSFER PROTOCOL
S113
ANALYZE HTML DOCUMENT
S114
SET SCREEN LAYOUT AND BACKGROUND COLOR FOR HTML DOCUMENT
S115
DETERMINE TYPES OF FONTS THAT CLIENT HAS
S116
CALL REGULAR FONT DETERMINING PROGRAM OF SERVER AND INFORM IT OF TYPES OF FONTS THAT CLIENT HAS
S117
A4

FIG. 16

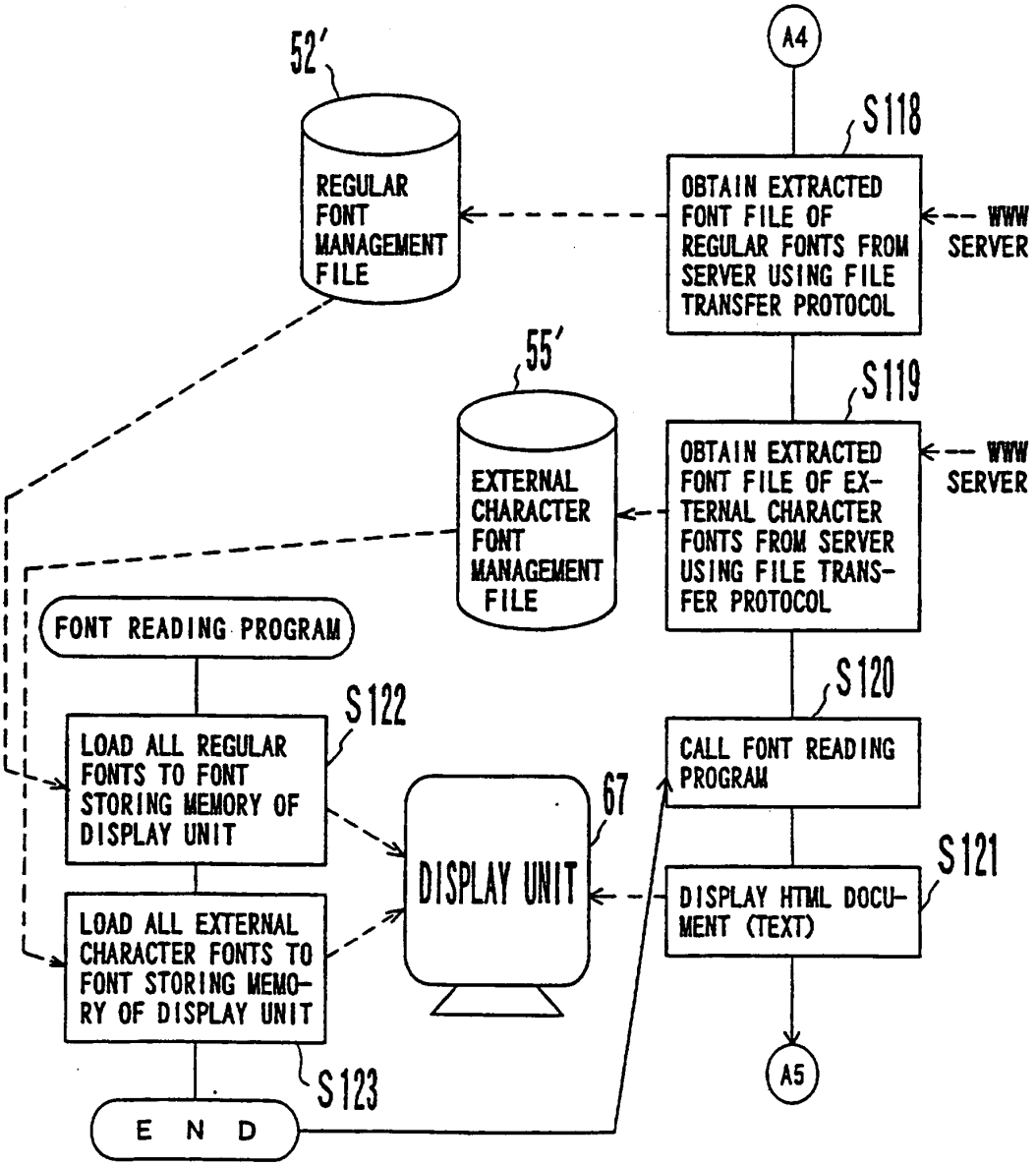
52'
REGULAR FONT MANAGEMENT FILE
A4
S118
OBTAIN EXTRACTED FONT FILE OF REGULAR FONTS FROM SERVER USING FILE TRANSFER PROTOCOL
WWW SERVER
55'
EXTERNAL CHARACTER FONT MANAGEMENT FILE
S119
OBTAIN EXTRACTED FONT FILE OF EXTERNAL CHARACTER FONTS FROM SERVER USING FILE TRANSFER PROTOCOL
WWW SERVER
FONT READING PROGRAM
S120
CALL FONT READING PROGRAM
S122
LOAD ALL REGULAR FONTS TO FONT STORING MEMORY OF DISPLAY UNIT
67
DISPLAY UNIT
S121
DISPLAY HTML DOCUMENT (TEXT)
LOAD ALL EXTERNAL CHARACTER FONTS TO FONT STORING MEMORY OF DISPLAY UNIT
A5
S123
E N D

FIG. 17

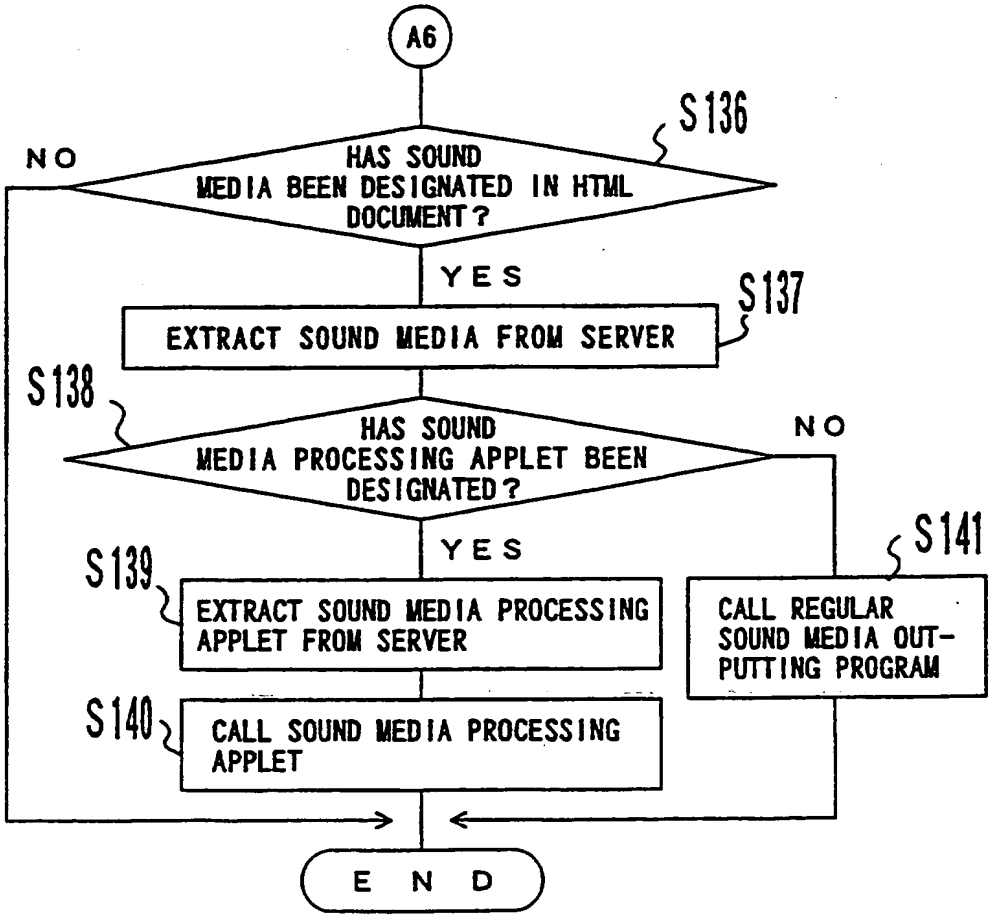
A6
S136
HAS SOUND MEDIA BEEN DESIGNATED IN HTML DOCUMENT?
NO
YES
S137
EXTRACT SOUND MEDIA FROM SERVER
S138
HAS SOUND MEDIA PROCESSING APPLET BEEN DESIGNATED?
NO
YES
S139
EXTRACT SOUND MEDIA PROCESSING APPLET FROM SERVER
S140
CALL SOUND MEDIA PROCESSING APPLET
S141
CALL REGULAR SOUND MEDIA OUTPUTTING PROGRAM
END

FIG. 19

Font Processing Apparatus in Network Environment and Method thereof

The present invention relates to a font processing apparatus for outputting characters with proper fonts through a communication network in a network environment, where a plurality of information processing units are connected to the network, and a method thereof.

In recent years, Internet services using WWW (World Wide Webs) have rapidly become more common. Companies, end users, and so forth have participated a wide variety of services such as information publications, commodity presentations, and commodity sales in a worldwide network environment. The WWWs represent Internet resources of information that server computers connected to the Internet provide. In the WWWs, a software tool termed a web browser is installed on a client computer that requests services to servers. The web browser communicates with the servers.

It is said that around 80% of companies, in the United States, have constructed local systems using the Internet services (namely, internal webs). An internal web is a kind of system that prohibits unauthorized users using a firewall or the like. Such a network is also referred to as an Intranet.

A large number of companies have entered the Internet businesses. Such companies establish their home pages on the Internet so as to provide a variety of services such as product presentations, sales, and maintenance services.

In addition, many companies are developing encrypting technologies for electronic commerce (EC). It is supposed that when such technologies are established, the number of business transactions using the Internet will be increased.

Further, the development methods of the application programs that run on the Internet have changed. Many innovations in development methods and providing methods thereof have resulted from the advent of new development languages such as Java.

Java is an interpreter, a network-based programming language. A program written in Java can be executed under any operating system (OS). A program written in Java is provided in the following

manner.

The server side has programs that run on the client side as applets. An applet is a client program that is coded in Java. The applet is provided from a network server (downloaded therefrom) and executed on the client side. This program is downloaded to the memory of the computer of the client. However, the applet is overwritten with another program after it is executed. Thus, applets are referred to as disposable software.

A Java-oriented browser on the client side downloads a required object (data or contents) and an applet for processing the object and executes the applet. A Java-oriented browser is a browser that can download an applet developed in Java and execute it. An example of Java-oriented browser is called Hot Java.

A major feature of this method is in that network servers prepare information (data) of inventory, price, and so forth necessary for transactions, and programs necessary for sales contracts, and they can control operations of the programs from the server side. It is considered that such a new providing method is an ideal environment for transactions (business) with end users connected to the Internet.

3NSDOCID: <GB 2316778A I >

Thus, the end users can connect their terminals to any sales services on the Internet without the need to construct special environments. In other words, when an end user connects his or her terminal to a particular service, he or she does not need to purchase a new software tool and install it on his/her terminal or add a new unit thereto.

Such a providing method is going to affect computer hardware. The terminal of each end user only requires a browser that accesses the network and a minimum OS that operates the browser. Other programs that each end user wants can be downloaded from the servers on the network. Thus, the memory, programs stored on the hard disc, and peripheral units necessary for the terminal can be minimized. As examples of such a terminal, so-called $500 terminals and portable terminals called PDAs (Personal Digital Assistants) are known.

Such a network environment can be treated as a system environment, where the whole network can be regarded as one OS and software tools that run on client terminals are provided by servers on the network.

So far, various Japanese language processing software products that run on main frame computers,

UNIX systems, personal computers, and so forth have been developed. They include software tools that perform font related processes and user defined character (external character) processes. In these systems, each system has an independent environment called an OS. A Japanese language processing software tool is installed in the OS.

Fig. 1A is a schematic diagram showing a conventional Japanese language processing system accomplished on a personal computer. In the system shown in Fig. 1A, various software products 1 are conventional software products that output media data such as characters, images, charts, and so forth. The various software products 1, for example, include business application programs that are used by companies for selling products.

Character information included in the various media that are output are, for example, font names, codes of character strings, character sizes, and character pitches. When necessary, a media data file may be created for output.

A terminal accessing driver 3 is a program that causes a terminal display 5 of the personal computer to display various media data. The terminal accessing driver 3 includes a font driver 4. The

3NSDOCID: <GB___2316778A__I_>

font driver 4 obtains a relevant font from a font file and displays it on the terminal display 5.

There are regular fonts in the system, such as a Mincho-tai font 6, a Gothic font 7, and a Mohitsu-tai font 8. In addition, other fonts can be optionally added.

However, such a Japanese language process is performed closedly in a single system such as a personal computer. Thus, required fonts should be added by the end user.

Fig. 1B is a schematic diagram showing a flow of processes performed between a server that is an information provider and a client that is a user terminal, in the case that the client receives various information services on the current WWW. The processes are performed in the following manner.

(a) Using a URL (Uniform Resource Locator), the user designates the location of a desired information service (home page) on a web browser 17 displayed on a display unit 23 of a WWW client 16. A URL is identification information that represents resources on the network in a uniform mannner.

(b) The web browser 17 obtains an IP (Internet Protocol) address corresponding to the designated URL, connects the WWW client 16 to the relevant

machine (WWW server) 11 on the network corresponding to the IP address, and downloads data of the "WWW server" home page to the WWW client 16.

The data of the home page has the format of an HTML document written in HTML (Hyper Text Markup Language). An HTML document is a kind of hyper text that contains links to other files. In this example, this data corresponds to document media 13.

In the WWW server 11, media to be provided 12 include documents, images, and sounds. The media 12 are categorized as three files that are the document media 13, image media 14, and sound media 15. The image media 14 and the sound media 15 are linked to the document media 13. In the HTML document, locations of various media data that are output along with a document have been defined by the links.

(c) The web browser 17 analyzes the document media 13 and obtains URLs of image media data and sound media data linked thereto.

(d) The web browser 17 obtains the IP address corresponding to the URLs of each media data, connects the WWW client 16 to the relevant machine on the network, and downloads the media data 14 and 15 to the WWW client 16.

(e) The web browser 17 displays and outputs the

downloaded media data 14 and 15 to the display unit 23 along with the document media 13. Normally, outputting programs corresponding to individual media are provided. A document outputting program 20 causes a character string in the document media 13 to be displayed using a font that is extracted from a font file 18 by a font reading program 19. The font reading program 19 corresponds to the font driver 4 shown in Fig. 1A. An image outputting program 21 displays data of the image media 14. A sound outputting program 22 outputs data of the sound media 15.

The conventional information processing systems perform one of the following font related processes.

(1) A system determines font resources required to print character media and supplies to a printer fonts that the printer does not have.

(2) Font resources for outputting character media are downloaded from a host computer or a server to a client beforehand. The client outputs character strings using the fonts.

(3) When a client requires a particular font, it requests that font to a host computer or a server.

However, in the method (1), the system should check the fonts that the printer has. In the method

BNSDOCID: <GB 2316778A I >

(2), all required fonts are not always downloaded. In the method (3), the client terminal should have a dedicated software tool for requesting a font.

In the Internet environment such as a WWW, since the entire network is treated as one OS, the Japanese language process that includes the font related process and the user defined character handling process should be performed in a new method.

For example, when a company provides services through home pages, various kinds of fonts (styles of type) will be used to create sophisticated screen displays. However, it is difficult for the terminal of each end user to have all the required fonts. Thus, a system that supplies such fonts to each user terminal and allows them to be displayed on the display unit thereof is required.

In addition, any UNIX systems and personal computers that run on Japanese OSs can handle JIS (Japanese Industrial Standard) class 1 characters and JIS class 2 characters. When a user terminal handles addresses and names of customers in Japanese, since the JIS class 1 characters and JIS class 2 characters are not sufficient to express them, a system that can handle user defined characters as external characters should be used.

RNSDOCID: <GB 2316778A I >

However, the external characters are defined by individual users. Thus, the external characters vary user by user. Consequently, external characters that one particular server has may not be displayed on other user terminals. On the Internet, a system that allows character strings including external characters to be displayed on each user terminal is desired.

Summary of the Invention

Embodiments of the present invention provide a font processing apparatus that allows various character expressions that a service provider wants to provide to be displayed on a user terminal regardless of font resources thereof in a network environment and a method thereof.

A font processing system embodying the present invention is used in a network environment, where a server unit that provides information and a client unit that receives the information are connected with a communication network. The font processing system includes a font processing unit in the server unit and a font processing unit in the client unit.

The font processing unit in the server unit has a storing unit and an outputting unit. The storing

3NSDOCID: <GB___2316778A__I_>

unit stores font resources and character media data. The font resources are required to output the character media data. The font resources are dynamically created correspondingly to the character media data. The outputting unit sends the character media data and the required font resources to the network.

The font processing unit in the client unit has an inputting unit and an outputting unit. The inputting unit receives the character media data and the required font resources from the network. The outputting unit outputs the character media data using the required font resources.

When the character media data is output to the client unit, the server unit dynamically extracts font data corresponding to the character media data from font resources of various styles of type that have been prepared beforehand, and creates the required font resources. The font resources are sent to the client unit through the network and output from the outputting unit of the client unit.

According to such a font processing system, font data used for the outputting process is provided through the network along with character media data to be output to the display unit of the client unit.

BNSDOCID: <GB 2316778A_I_>

Thus, the operating environment relating to the character display of each user terminal in the network environment is dynamically prepared and supplied by the server unit.

The present invention will become more apparent in light of the following detailed description of an embodiment thereof, as illustrated in the accompanying drawings.

Brief Description of the Drawings

Fig. 1A is a schematic diagram showing a conventional Japanese language processing system;

Fig. 1B is a schematic diagram showing a structure of the Internet;

Fig. 2A is a schematic diagram showing the principle of a font processing system embodying the present invention;

Fig. 2B is a schematic diagram showing a font processing system in the Internet environment;

Fig. 3 is a schematic diagram showing the structure of an information processing apparatus;

Fig. 4 is a program list showing applets designations in an HTML document;

Fig. 5 is a schematic diagram showing the format

of an extracted font file;

Fig. 6 is a schematic diagram showing font data for characters;

Fig. 7 is the first part of a flow chart showing a first browser process;

Fig. 8 is the second part of a flow chart showing the first browser process;

Fig. 9 is the third part of a flow chart showing the first browser process;

Fig. 10 is a flow chart showing the process of a character resource extracting program;

Fig. 11 is the first part of a flow chart showing the process of a regular font type determining program/external character font type determining program;

Fig. 12 is the second part of a flow chart showing the process of the regular font type determining program/external character font type determining program;

Fig. 13 shows flow charts showing the processes of a document media outputting program/font reading program;

Fig. 14 shows flow charts showing the processes of an image media processing program/sound media processing program;

3NSDOCID: <GB 2316778A I >

Fig. 15 shows flow charts showing the processes of the media outputting programs;

Fig. 16 is the first part of a flow chart showing a second browser process;

Fig. 17 is the second part of a flow chart showing the second browser process;

Fig. 18 is the third part of a flow chart showing the second browser process; and

Fig. 19 is the fourth part of a flow chart showing the second browser process.

Next, with reference to the accompanying drawings, embodiments of the present invention will be described in detail.

Fig. 2A is a schematic diagram showing the principle of a font processing system according to an embodiment. The font processing system shown in Fig. 2A is used in a network environment, where a server unit 31 that provides information and a client unit 35 that receives the information therefrom are connected with a communication network 34. The server unit 31 has a font processing unit. Likewise, the client unit 35 has a font processing unit.

The server unit 31 is, for example, an information processing unit of a service provider that provides information such as services on the Internet. The client unit 35 is, for example, an end user terminal unit that receives services on the Internet.

The font processing unit in the server unit 31 has a storing unit 32 and an outputting unit 33.

The storing unit 32 stores font resources and character media data. The font resources are required to output the character media data. The font resources are dynamically created correspondingly to the character media data.

The outputting unit 33 sends the character media data and the required font resources to the network 34.

The font processing unit in the client unit 35 has an inputting unit 36 and an outputting unit 37.

The inputting unit 36 receives the character media data and the required font resources from the network 34.

The outputting unit 37 outputs the character media data on a display unit or the like using the required font resources.

When the server unit 31 outputs the character media data to the client unit 35, the server unit 31

dynamically extracts font data corresponding to the character media data from font resources of various styles of type and creates the required font resources. The created font resources are sent to the client unit 35 through the network 34 and then output from the outputting unit 37.

According to the font processing system, along with character media data to be output to the display unit of the client unit 35 or the like, font data used for the outputting process thereof is provided through the network 34. Thus, the operating environment of a character display on each end user terminal is dynamically prepared and supplied on the server side in the network environment.

When necessary, font data for outputting characters is supplied from the server side. Thus, since the user terminal does not need to prepare such font data, the structure and function of the user terminal can be simplified. In addition, since the service provider designates fonts for character media, it can create output screens in a combination of the fonts. Thus, the service provider can provide information with various character expressions.

For example, the storing unit 32 shown in Fig. 2A corresponds to a memory 72 or an external storing

unit 75 shown in Fig. 3, which is explained later. The outputting unit 33 and the inputting unit 36 correspond to a network connecting unit 77 shown in Fig. 3. The outputting unit 37 corresponds to an outputting unit 74 shown in Fig. 3 or the media outputting program 64, 65, 66 and the display unit 67 shown in Fig. 2B, which is also explained later.

When each company provides services such as product presentations and product sales using the Internet, on such as a WWW, a server unit of the company provides each requesting end user terminal with information such as characters, still pictures, moving pictures and sounds that are displayed and output to the end user terminal. When character media are displayed, the provider wants to use various fonts to improve the expressions of data regardless of the environment of each end user terminal.

To satisfy such a demand, in addition to the above-described media data, the server unit supplies font resources required to output the media and a program part that performs the outputting process thereof in the present invention.

In the conventional processing method shown in Fig. 1B, each end user should prepare fonts and

PNSDOCID: <GB 2316778A I >

output programs beforehand at the terminal side. On the other hand, according to the present invention, with the object-oriented concept, along with the media, fonts and output programs are supplied. The feature of the present invention is in that the operating environments of individual end user terminals connected to network environments are prepared and provided by the server.

Fig. 2B is a schematic diagram showing a font processing system that accomplishes a font related process on the Internet. The font processing system shown in Fig. 2B comprises a WWW server 41 and a WWW client 61. The WWW server 41 provides information. The WWW client 61 is connected to the WWW server 41 by a communication network.

The WWW server 41 has document media 43, image media 44, and sound media 45 that are files that store media to be provided 42. In addition, the WWW server 41 has a character resources extracting program 46, a document media outputting program 40, a document media processing program 47, an image media processing program 48, a sound media processing program 49, a regular font type determining program 51, an external character font type determining program 54, and a font reading program 56.

As with the document media 13 shown in Fig. 1B, the document media 43 is written in HTML. The document media 43 includes character media such as character strings and has links to the image media 44 and the sound media 45. The regular font type determining program 51 and the external character font type determining program 54 are programs written in, for example, C language and executed on the WWW server 41.

Further, the character resources extracting program 46, the document media outputting program 40, the document media processing program 47, the image media processing program 48, the sound media processing program 49, and the font reading program 56 are applet programs written in, for example, the Java language. These programs are downloaded to the WWW client 61 and then executed.

The character resource extracting program 46, the document media outputting program 40, and the font reading program 56 are common programs prepared beforehand in the system. On the other hand, the document media processing program 47, the image media processing program 48, and the sound media processing program 49 are programs that the service provider can individually designate.

The WWW client 61 has a display unit 67. In addition, the WWW client 61 has a web browser 62, a regular font reading program 63, a regular document media outputting program 64, a regular image media outputting program 65, and a regular sound media outputting program 66.

The web browser 62 is a Java-oriented browser that downloads applets from the WWW server 41 and executes them. The programs 63, 64, 65, and 66 are output processing programs in the WWW client 61 and correspond to, respectively, the font reading program 19, the document outputting program 20, the image outputting program 21, and the sound outputting program 22 shown in Fig. 1B.

The font processing system shown in Fig. 2B operates as follows.

(a) The user designates the location of a desired home page with the web browser 62 on the display unit 67 using a URL or the like.

(b) The web browser 62 obtains an IP address corresponding to the designated URL, connects the WWW client 61 to the WWW server 41 that is the relevant server corresponding to the IP address on the network, and downloads to the WWW client 61 the document media 43 corresponding to the home page and

the character resources extracting program 46 to which the document media 43 is linked.

(c) The web browser 62 analyzes the document media 43, obtains URLs of image media data, sound media data, and so forth linked thereto, and calls the character resources extracting program 46. The character resources extracting program 46 downloads font resources required to output and display an HTML document, the document media outputting program 40, and the font reading program 56 from the WWW server 41 to the WWW client 61. The processing of the character resources extracting program 46 is composed of the following two steps (X) and (Y).

(X) The character resources extracting program 46 calls the regular font type determining program 51 in the WWW server 41 using a protocol such as Java so as to extract required regular fonts from the regular font management file 50. At this point, fonts corresponding to characters used in the document media 43 are extracted and stored in an extracted font file 52 of regular fonts.

The regular font type determining program 51 calls the external character font type determining program 54 so as to extract required external character fonts from an external character font management file 53.

EP 0 831 677 A1

At this point, fonts corresponding to external characters used in the document media 43 are extracted and stored in an extracted font file 55 of external character fonts.

(Y) The character resource extracting program 46 downloads the extracted font files 52, 55, the document media outputting program 40, and the font reading program 56 from the WWW server 41 to the WWW client 61. The files 52 and 55 are stored as a regular font management file 52' and an external character font management file 55' in the WWW client 61.

(d) The web browser 62 obtains the IP addresses corresponding to the URLs of individual media data obtained in (c), connects the WWW client 61 to the relevant servers on the network, and downloads the image media 44 and the sound media 45 therefrom.

(e) The web browser 62 outputs and displays the downloaded media data 44 and 45 to the display unit 67 along with the document media 43. Outputting programs corresponding to the individual media are provided.

When the document media 43 is output, the downloaded font reading program 56 and document media outputting program 40 are used instead of the regular

font reading program 63 and the regular document media outputting program 64.

The document media outputting program 40 is a program for displaying character strings in the document media 43. The document media outputting program 40 causes the font reading program 56 to extract fonts. The font reading program 56 extracts relevant fonts for individual characters from the regular font management file 52' and the external character font management file 55' and supplies the extracted fonts to the document media outputting program 40.

The regular image media outputting program 65 displays data of the image media 44. The regular sound media outputting program 66 outputs data of the sound media 45.

As with the image media 44 and the sound media 45, other document media (not shown) can be linked to the document media 43 in the WWW server 41. In (d), the document media processing program 47, the image media processing program 48, and the sound media processing program 49 can be downloaded from the WWW server 41.

In this case, in (e), the document media processing program 47 displays data of document media linked to the document media 43. The image media

PNSDOCID: <GB 2316777A I >

processing program 48 displays data of the image media 44. The sound media processing program 49 outputs data of sound media 45.

Thus, the service provider can designate the editing process and/or outputting process of the other document media, the image media 44, and the sound media 45, and cause the WWW client 61 to perform the processes.

According to the font processing method, the server (host computer) dynamically determines fonts needed by a client terminal that does not have the regular character media outputting program 64 and that needs to output character media, and dynamically downloads the determined fonts along with character media (output document) to the client (end user terminal). Thus, even if a terminal is simplified and does not have regular fonts, it can display various types of characters.

In addition, the server provides program portions (applets) that output and/or display fonts. The applets are downloaded to the client along with the character media. When the client has completed the outputting process, the media data, font resources, and program portions that have been downloaded from the server are deleted from the client. Thus, the

operating environment of each end user terminal on the network is provided and controlled by the server.

In Europe and America, a large number of fonts such as Times, Helvetica, and Courier are used. In addition, system fonts corresponding to various platforms have been used. Thus, the font processing system shown in Fig. 2B is effective to display fonts for other languages (non-Japanese languages). In particular, the combination of the character resources extracting program 46, the document media outputting program 40, the regular font type determining program 51, and the font reading program 56 is suitable for use in Europe and America as well as in Japan.

On the other hand, the combination of the character resources extracting program 46, the document media outputting program 40, the external character font type determining program 54, and the font reading program 56 is suitable for the use of external characters in Japan. The structure, in which such a combination and the regular font type determining program 51 exists, is suitable for the use of regular fonts and external character fonts in Japan.

Fig. 3 is a schematic diagram showing an example

of the structure of an information processing apparatus corresponding to the WWW server 41 or the WWW client 61 shown in Fig. 2B. The information processing apparatus shown in Fig. 3 comprises a CPU (Central Processing Unit) 71, a memory 72, an inputting unit 73, an outputting unit 74, an external storing unit 75, a medium driving unit 76, and a network connecting unit 77. These units are mutually connected with a bus 78.

The CPU 71 executes a program stored in the memory 72 and accomplishes each process of the WWW server 41 or the WWW client 61. The memory 72 is, for example, a ROM (Read Only Memory) and RAM (Random Access Memory).

The inputting unit 73 is equivalent to, for example, a keyboard and a pointing device. The outputting unit 74 is equivalent to, for example, a display unit 67 or a printer.

The external storing unit 75 is equivalent to a magnetic disc unit, an optical disc unit, or a magneto-optical disc unit. The external storing unit 75 can store programs and data. In addition, the external storing unit 75 can be used as a database that stores font resources.

The medium driving unit 76 drives a portable

storing medium 79 and accesses the stored contents thereon. The portable storing medium 79 is an arbitrary computer-readable storage medium such as a memory card, a floppy disc, a CD-ROM (Compact Disc Read Only Memory), an optical disc, or a magneto-optical disc, from which a computer can read programs and data. The portable storing medium 79 stores programs for performing processes of the WWW server 41 or the WWW client 61 along with data.

The network connecting unit 77 is connected to the WWW through a communication network such as a LAN (Local Area Network) and exchanges data necessary for communications. The WWW server 41 and the WWW client 61 can receive required data and programs from the network through the respective network connecting units 77.

It should be noted that the WWW server 41 and the WWW client 61 do not need to have all the structural elements shown in Fig. 3.

Next, with reference to Figs. 4 to 6, the structure of a file used in the font processing system shown in Fig. 2B will be described.

Fig. 4 is a program list showing applets in an HTML document of the document media 43. In the HTML document shown in Fig. 4, line L1 includes the file

name of an applet corresponding to the character resources extracting program 46. Line L2 contains the URL of the HTML document.

Line L3 contains the location of the applet (namely, the address of the server). Line L4 contains the file name of an applet corresponding to the document media processing program 47. Line L5 designates character media.

Line L6 contains the file name of an applet corresponding to the image media processing program 48. Line L7 designates image media and contains the file name of the image media 44.

Line L8 contains the file name of an applet corresponding to the sound media processing program 49. Line L9 designates the sound media and contains the file name of the sound media 45.

In this embodiment, since the file name of the document media outputting program 40 is stored in the character resources extracting program 46 beforehand, it is not necessary to contain the file name thereof in the HTML document.

The web browser 62 analyzes the HTML document, extracts individual media data and the files of the applets, and downloads them to the client side.

Fig. 5 is a schematic diagram showing an example

of the format of the extracted font files 52 and 55. In the file shown in Fig. 5, character font data is categorized as fonts such as Mincho-tai font, Gothic font, and Mohitsu-tai font. Each font data has a structure as shown in Fig. 6.

In Fig. 6, each font name is followed by font data. The font data is composed of character code, character size, and character pattern. The character code is identification information that identifies the character type. The character size represents the size of the characters. The character pattern is, for example, bit map data and vector data that represent the shape of characters that are output.

Next, with reference to Figs. 7 to 15, processes performed in the font processing system shown in Fig. 2B, in the case that applets written in Java are used, will be described.

Figs. 7, 8, and 9 are flow charts showing processes performed by the web browser 62. When the web browser 62 starts the process shown in Fig. 7, it reads and analyzes a URL that has been input (at step S1) and connects the WWW client 61 to the relevant server 41 corresponding to the URL (at step S2).

Next, the web browser 62 extracts a HTML document corresponding to the document media 43 from the

server 41 with a file transfer protocol (FTP) (at step S3), analyzes it (at step S4), and sets the screen layout and background color for the HTML document (at step S5).

Next, the web browser 62 determines whether or not the character resources extracting applet 46 has been designated in the HTML document (at step S6). When the determined result is Yes, the web browser 62 extracts the character resources extracting applet 46 from the server 41 (at step S7) and executes it (at step S8).

Thereafter, the web browser 62 extracts the document media outputting applet 40 specified in the character resources extracting applet 46 from the server 41, executes it, and displays the text portion (home page) of the HTML document (at step S9). When the determined result at step S6 is No, the web browser 62 calls the regular document media outputting program 64 and displays the text portion of the HTML document (at step S10).

Next, the web browser 62 determines whether or not document media has been designated in the HTML document (at step S11 in Fig. 8). When the determined result is Yes, the web browser 62 extracts the document media from the server 41 (at step S12).

Thereafter, the web browser 62 determines whether or not the document media processing applet 47 has been designated (at step S13). When the determined result at step S11 is No, the flow advances to step S17.

When the determined result at step S13 is Yes, the web browser 62 extracts the document media processing applet 47 from the server 41 (at step S14) and executes it (at step S15). When the determined result at step S13 is No, the web browser 62 executes the regular document media outputting program 64 (at step S16).

Next, the web browser 62 determines whether or not image media has been designated in the HTML document (at step S17). When the determined result is Yes, the web browser 62 extracts the image media 44 from the server 41 (at step S18). Next, the web browser 62 determines whether or not the image media processing applet 48 has been designated (at step S19). When the determined result at step S17 is No, the flow advances to step S23 shown in Fig. 9.

When the determined result at step S19 is Yes, the web browser 62 extracts the image media processing applet 48 from the server 41 (at step S20) and executes it (at step S21). When the determined result at step S19 is No, the web browser 62 executes

INSDOCID: <GB 2316778A I >

the regular image media outputting program 65 (at step S22).

Thereafter, the web browser 62 determines whether or not sound media has been designated in the HTML document (at step S23 in Fig. 9). When the determined result at step S23 is Yes, the web browser 62 extracts the sound media 45 from the server 41 (at step S24). Thereafter, the web browser 62 determines whether or not the sound media processing applet 49 has been designated (at step S25). When the determined result at step S23 is No, the web browser 62 completes the process.

When the determined result at step S25 is Yes, the web browser 62 extracts the sound media processing applet 49 from the server 41 (at step S26) and executes it (at step S27). Thereafter, the web browser 62 completes the process. When the determined result at step S25 is No, the web browser 62 executes the regular sound media outputting program 66 (at step S28) and completes the process.

Fig. 10 is a flow chart showing the process of the character resources extracting applet 46 called at step S8 shown in Fig. 7. The character resources extracting applet 46 includes file names of the regular font type determining program 51, the

document media outputting applet 40, and the font reading applet 56.

When the character resources extracting applet 46 starts the process shown in Fig. 10, it receives the IP address of the server 41 and the HTML document name of the document media 43 from the web browser 62 (at step S31) and connects the WWW client 61 to the server 41 (at step S32).

Thereafter, the character resources extracting applet 46 determines the types of fonts that the client 61 has (at step S33), calls the regular font type determining program 51 of the server 41, and informs the server 41 of the types of fonts that the client 61 has (at step S34).

Next, the character resources extracting applet 46 downloads the extracted font file 52 of regular fonts to be used using FTP from the server 41 and stores it as the regular font management file 52' in the client 61 (at step S35). Likewise, the character resource extracting applet 46 downloads the extracted font file 55 of external character fonts from the server 41 and stores it as the external character font management file 55' in the client 61 (at step S36).

Thereafter, the character resources extracting applet 46 downloads the document media outputting

BNSDOCID: <GB 2316778A | >

applet 40 from the server 41 using FTP (at step S37) and then the font reading applet 56 (at step S38). Thereafter, the character resources extracting applet 46 completes the process.

Thus, the character resources extracting applet 46 downloads the regular font management file 52', the external character font management file 55', the document media outputting applet 40, and the font reading applet 56 to the client 61. Thus, the client 61 can output the fonts that the information provider has designated.

Figs. 11 and 12 are flow charts showing processes of the regular font type determining program 51 called at step S34 shown in Fig. 10 and the external character font type determining program 54 called thereby. When the regular font type determining program 51 starts the process shown in Fig. 11, it receives the HTML document name from the character resources extracting applet 46 that has called the regular font type determining program 51 and reads the relevant HTML document (the document media 43) corresponding thereto (at step S41).

Next, the regular font type determining program 51 clears the contents of an external character code cumulative table 81 and a regular character code

cumulative table 82 stored in the work region of the memory of the server 41 (at step S42). In addition, the regular font type determining program 51 resets the character extracting counter to zero (at step S43).

The regular font type determining program 51 extracts one character to be processed from the HTML document (at step S44) and determines whether or not the character is an external character (at step S45). Normally, the external character codes have values out of the range of values of the regular character codes. Thus, when the regular font type determining program 51 reads a character code, it can determine whether or not it is an external character.

When the extracted character is an external character, the determined result is Yes at step S45, the regular font type determining program 51 calls the external character font type determining program 54 in the server 41 (at step S46). When it is not an external character, the regular font type determining program 51 determines whether or not the font of the character corresponds to any of the types of fonts in the client 61 informed of by the character resources extracting applet 46 (at step S47). When the determined result is Yes, the flow advances to step

RNSDOCID: <GB 2316778A I >

S52.

When the determined result at step S47 is No, the regular font type determining program 51 determines whether or not the extracted character has been registered in the regular character code cumulative table 82 (at step S48). When the determined result is Yes, the flow advances to step S52. When the determined result is No, the regular font type determining program 51 registers the character code as a new entry of a font in the regular character code cumulative table 82 (at step S49).

Thereafter, the regular font type determining program 51 increments the value of a character extracting counter by one (at step S52). Next, the flow advances to step S53. At step S53, the regular font type determining program 51 determines whether or not there is a character left to be processed. When the determined result is No, the flow returns to step S44. Thus, the regular character code cumulative table 82 cumulates different character codes for each font.

The external character font type determining program 54 determines whether or not a character to be processed has been registered in the external character code cumulative table 81 (at step S50).

When the determined result is Yes, the external character font type determining program 54 completes the process. When the determined result is No, the external character font type determining program 54 registers the external character code as a new entry of a font in the external character code cumulating table 81 (at step S51) and completes the process.

When the external character font type determining program 54 completes the process, the regular font type determining program 51 performs the process after step S52. Thus, different external character codes for each font are cumulated in the external character code cumulative table 81.

Next, the regular font type determining program 51 extracts fonts corresponding to characters registered in the regular character code cumulative table 82 from the regular font management file 50 in the server 41 and creates the extracted font file 52 of regular fonts (at step S54 shown in Fig. 12).

The regular font type determining program 51 extracts fonts corresponding to the external characters registered in the external character code cumulative table 81 from the external character management file 53 and generates the extracted font file 55 of external character fonts (at step S55) and

completes the process.

Thus, only fonts necessary for displaying the document media 43 are selected from the font management files 50 and 53 and stored in the files 52 and 55, respectively. These files 52 and 55 are downloaded to the client 61 as the regular font management file 52' and the external character font management file 55'.

Fig. 13 shows flow charts showing processes of the document media outputting applet 40 called at step S9 shown in Fig. 7 and the font reading applet 56 called thereby. When the document media outputting applet 40 starts the process shown in Fig. 13, it calls the font reading applet 56 in the client 61 (at step S61).

The font reading applet 56 loads all regular fonts in the regular font management file 52' to the font storing memory in the display unit 67 (at step S71). In addition, the font reading applet 56 loads all external character fonts in the external character font management file 55' to the font storing memory (at step S72). Thereafter, the font reading applet 56 completes the process.

Next, the document media outputting applet 40 displays the document media 43 on the display unit 67

with the fonts stored in the font storing memory (at step S62) and then completes the process.

As with the document media outputting applet 40, the document media processing applet 47 called at step S15 shown in Fig. 8 causes the font reading applet 56 to display the document media 43.

Fig. 14 is flow charts showing processes of the image media processing applet 48 called at step S21 shown in Fig. 8 and the sound media processing applet 49 called at step S27 shown in Fig. 9.

The image media processing applet 48 melts (expands) compressed data of the image media 44 (at step S81) and edits the resultant data corresponding to the method written in the applet 48 (at step S82). Thereafter, the image media processing applet 48 causes the display unit 67 to display the edited data (at step S83) and completes the process.

The sound media processing applet 49 melts compressed data of the sound media 45 (at step S91) and edits the resultant data corresponding to the method written in the applet 49 (at step S92). Thereafter, the sound media processing applet 49 outputs the edited data to a speaker of the display unit 67 (at step S93) and then completes the process.

Fig. 15 is a flow chart showing processes of the

regular document media outputting program 64 called at steps S10 shown in Fig. 7 and S16 shown in Fig. 8, the regular image media outputting program 65 called at step S22 shown in Fig. 8, and the regular sound media outputting program 66 called at step S28 shown in Fig. 9.

The document media outputting program 64 causes the regular font reading program 63 to display the document media 43 on the display unit 67 (at step S101) and then completes the process.

The image media outputting program 65 melts compressed data of the image media 44 (at step S102), displays the resultant data on the display unit 67 (at step S103), and completes the process.

The sound media outputting program 66 melts compressed data of the sound media 45 (at step S104), outputs the resultant data to the speaker of the display unit 67 (at step S105), and then completes the process.

In the above-described embodiment, the character resources extracting program 46, the document media outputting program 40, and the font reading program 56 are downloaded from the server 41. However, these programs can be disposed in the client 61.

Figs. 16, 17, 18, and 19 are flow charts showing

processes of the web browser 62 and the font reading program 56 in the case that the web browser 62 has functions of the character resource extracting program 46 and the document media outputting program 40 and that the client 61 has the font reading program 56.

When the web browser 62 starts the process shown in Fig. 16, it extracts and analyzes an URL being input (at step S111) and connects the client to the relevant server 41 (at step S112).

Thereafter, the web browser 62 extracts an HTML document corresponding to the document media 43 from the server 41 with the FTP (at step S113), analyzes it (at step S114), and sets the screen layout and background color for the HTML document (at step S115).

Thereafter, the web browser 62 determines the types of fonts that the client 61 has (at step S116), calls the regular font type determining program 51 of the server 41, and informs it of the types of fonts that the client 61 has (at step S117).

Next, the web browser 62 downloads the file 52 from the server 41 using the FTP and stores it as the regular font management file 52' in the client 61 (at step S118 shown in Fig. 17). Likewise, the web

browser 62 downloads the file 55 from the server 41, stores it as the external character font management file 55' in the client 61 (at step S119). Thereafter, the web browser 62 calls the font reading program 56 (at step S120).

The font reading program 56 loads all the fonts in the regular font management file 52' to the font storing memory of the display unit 67 (at step S122). In addition, the font reading program 56 loads all the external character fonts in the external character management file 55' to the font storing memory (at step S123). Thereafter, the font reading program 56 completes the process.

Next, the web browser 62 displays the text of the HTML document on the display unit 67 using the fonts stored in the font storing memory (at step S121).

Next, the web browser 62 determines whether or not document media has been designated in the HTML document (at step S124 shown in Fig. 18). When the determined result is Yes, the web browser 62 extracts the document media from the server 41 (at step S125). Thereafter, the web browser 62 determines whether or not the document media processing applet 47 has been designated (at step S126). When the determined result at step S124 is No, the flow advances to step

S130.

Next, when the determined result at step S126 is Yes, the web browser 62 extracts the document media processing applet 47 from the server 41 (at step S127) and executes it (at step S128). When the determined result at step S126 is No, the web browser 62 executes the regular document media outputting program 64 (at step S129). Thereafter the flow advances to step S130.

Next, the web browser 62 determines whether or not image media has been designated in the HTML document (at step S130). When the determined result at step S130 is Yes, the web browser 62 extracts the image media 44 from the server 41 (at step S131). Thereafter, the web browser 62 determines whether or not the image media processing applet 48 has been designated (at step S132). When the determined result at step S130 is No, the flow advances to step S136, shown in Fig. 19.

When the determined result at step S132 is Yes, the web browser 62 extracts the image media processing applet 48 from the server 41 (at step S133) and executes it (at step S134). When the determined result at step S132 is No, the web browser 62 executes the regular image media outputting

program 65 (at step S135).

Next, the web browser 62 determines whether or not sound media has been designated in the HTML document (at step S136 in Fig. 19). When the determined result at step S136 is Yes, the web browser 62 extracts the sound media 45 from the server 41 (at step S137). Thereafter, the web browser 62 determines whether or not the sound media processing applet 49 has been designated (at step S138). When the determined result at step S136 is No, the web browser 62 completes the process.

When the determined result at step S138 is Yes, the web browser 62 extracts the sound media processing applet 49 from the server 41 (at step S139) and executes it (at step S140). Thereafter, the web browser 62 completes the process. When the determined result at step S138 is No, the web browser 62 executes the regular sound media outputting program 66 (at step S141) and completes the process.

In the font processing system shown in Fig. 2B, the extracted font files 52 and 55 are separately created and downloaded to the client 61. However, the files 52 and 55 can be combined as one extracted font file and then downloaded to the client 61. In this case, character codes of regular characters and

external characters are stored in the same file. However, according to the range of character code values and so forth, the regular characters and external characters can be distinguished.

In embodiments, in a network environment such as the Internet, the server side can control information, programs, and so forth that it provides, so as to improve expressions in character media.

In particular, once the server side prepares fonts, each end user terminal can display them. Thus, each end user terminal needs only a minimum number of software tools. Consequently, the cost of the end user terminal can be reduced.

In addition to JIS class 1 characters and JIS class 2 characters, each end user terminal can display other external characters that service providers define and add.

Although the present invention has been shown and described with respect to a preferred embodiment thereof, it should be understood by those skilled in the art that the foregoing and various other changes, omissions, and additions in the form and detail thereof may be made therein without departing from the spirit and scope of the present invention.

CLAIMS

1. A font processing apparatus for a server unit (13, 41) that is an information processing unit on a communication network (34) and provides information, comprising:

storing means (32) storing font resources and character media data, the font resources being required to output the character media data,

means to dynamically create the font resources corresponding to the character media data; and

outputting means (33) arranged to send the character media data and the required font resources to the network.

2. The font processing apparatus as set forth in claim 1, wherein:

the means to create the font resources is a character type extracting means (51, 54) adapted to extract the required font resources from font resources of a plurality of styles of type that have been prepared in advance,

said storing means (32) stores document data including the character media data, and

said character type extracting means (51, 54) analyzes the document data, extracts font data

corresponding to each character of the character media data from the font resources of the plurality of styles, and creates the required font resources including the extracted font data.

3. The font processing apparatus as set forth in claim 2,

wherein said character type extracting means (51, 54) excludes font data stored in a client unit, that is an information processing unit and has received the character media data, from the required font resources.

4. The font processing apparatus as set forth in claim 1, 2, or 3,

wherein said storing means (32) also stores a character resources extracting program (46) for causing a client unit (35, 61), that is an information processing unit and has received the character media data, to download the required font resources from the server unit (31, 41),

wherein said outputting means (33) sends the character resources extracting program (46) to the network (34) along with the character media data, to enable the client unit (35, 61) to output the

character media data using the character resources extracting program.

5. The font processing apparatus as set forth in claim 4,

wherein said storing means (32) also stores a font reading program (56) for extracting font data for each character from the required font resources,

wherein the character resources extracting program (46) is adapted to run on the client unit (35, 61) to download the font reading program (56) from the server unit (31, 41) through the network (34), and to output the character media data using the font reading program (56).

6. The font processing apparatus as set forth in claim 5,

wherein said storing means (32) also stores a document media outputting program (40) for outputting the character media data using the font data extracted by the font reading program (56), adapted such that said client unit (35, 61) downloads the document media outputting program (40) from the server unit (31, 41) through the network (34) and outputs the character media data.

BNSDOCID: <GB 2316778A I >

7. The font processing apparatus as set forth in claim 6,

wherein said storing means (32) stores at least one of the character resources extracting program (46), the font reading program (56), and the document media outputting program (40) in the format of a program written in an interpreter, network-based language.

8. The font processing apparatus as set forth in any preceding claim,

wherein said storing means (32) stores the character media data including at least one external character arbitrarily defined.

9. The font processing apparatus as set forth in claim 8, further comprising:

external character extracting means (54) for extracting font data for the external character from font resources of a plurality of styles of external character that have been prepared in advance,

wherein said storing means (32) stores document data including the character media data, and

wherein said external character extracting means (54) analyzes the document data, extracts font data

BNSDOCID: <GB 2316778A I >

corresponding to the external character from the font resources of the plurality of styles, and creates the required font resources including the extracted font data.

10. A font processing apparatus for a server unit (31, 41) that is an information processing unit on a communication network (34) and provides information, comprising:

storing means (32) storing external character data corresponding to at least one external character arbitrarily defined and font resources required to output the external character; and

outputting means (33) arranged to send the external character data and the required font resources to the network.

11. A font processing apparatus for a client unit (35, 61) that is an information processing unit on a communication network and receives information, comprising:

inputting means (36) arranged to receive font resources and character media data from the network (34), the font resources being required to output the character media data, the font resources being dynamically

created corresponding to the character media data; and

outputting means (37) arranged to output the character media data using the required font resources.

12. The font processing apparatus as set forth in claim 11,

wherein said inputting means (36) receives from the network (34) the required font resources excluding font data that the client unit already has.

13. The font processing apparatus as set forth in claim 11 or 12, further comprising:

character resources extracting means (46) arranged to download the required font resources from a server unit (31, 41), that is an information processing unit and has sent the character media data, through the network (34).

14. The font processing apparatus as set forth in claim 13, further comprising:

font reading means (56, 63) arranged to extract font data for each character from the required font resources,

wherein said outputting means (37) outputs the

BNSDOCID: <GB 2316778A I >

character media data in cooperation with said font reading means (56, 63).

15. The font processing apparatus as set forth in claim 11, 12, 13, or 14,

wherein said inputting means (36) receives from the network (34) the character media data including at least one external character arbitrarily defined.

16. A font processing system for use in a network environment in which a server unit (31, 41) that provides information and a client unit (35, 61) that receives the information are connected with a communication network (34), comprising:

storing means (32) in the server unit (31,41) storing font resources and character media data,the font resources being required to output the character media data, the font resources being dynamically created correspondingly to the character media data;

first outputting means (33) in the server unit (31,41) arranged to send the character media data and the required font data to the network (34);

inputting means (36) in the client unit (35,61) arranged to receive the character media data and the required font resources from the network (34); and

PNSDOCID: <GB 2314778A I >

second outputting means (37) in the client unit arranged to output the character media data using the required font resources.

17. A resources supplying apparatus for a server unit (31, 41) that is an information processing unit on a communication network (34) and provides information, comprising:

storing means (32) storing information resources and media data, the information resources being required to output the media data, the required information resources being dynamically created correspondingly to the media data; and

outputting means (33) arranged to send the media data and the required information resources to the network (34).

18. A computer-readable storage medium (72, 75, 79), when used by a computer on a communication network (34) for providing information, used to direct the computer to perform the functions of:

dynamically creating font resources required to output character media data correspondingly to the character media data; and

sending the character media data and the required

BNSDOCID: <GB 2316778A I >

font resources to the network (34).

19. A computer-readable storage medium (72, 75, 79), when used by a computer on a communication network (34) for receiving information, used to direct the computer to perform the functions of:

receiving font resources and character media data from the network (34), the font resources being required to output the character media data, the required font resources being dynamically created correspondingly to the character media data, and

outputting the character media data using the required font resources.

20. A font processing method in a network environment in which a server unit (31, 34) that provides information and a client unit (35, 61) that receives the information are connected with a communication network (34), comprising the steps of:

dynamically creating font resources required to output character media data correspondingly to the character media data in the server unit (31, 41) ; and

sending the character media data and the required font resources from the server unit (31, 41) to the

client unit (35, 61) through the network (34).

21. The font processing method as set forth in claim 20,

wherein the character media data is output using the required font resources from the client unit (35, 61).

22. A resources providing method in a network environment in which a server unit (31, 41) that provides information and a client unit (35, 61) that receives the information are connected with a communication network (34), comprising the steps of:

dynamically creating information resources required to output media data correspondingly to the media data in the server unit (31, 41) ; and

sending the media data and the required information resources from the server unit (31, 41) to the client (35, 61) unit through the network (34).

BNSDOCID: <GB 2316778A I >

The Patent Office

SP

Application No: GB 9705647.7
Claims searched: 1 - 22

Examiner: Paul Nicholls
Date of search: 15 May 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): B6F (FAM), G4A (AFGDC), H4T (TDAA)

Int Cl (Ed.6): G06F 15/163, 17/21, 17/30

Other: Online: WPI, COMPUTER, BUSSOFT

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	GB 2,051,527 A (KWOK and LAI) - see page 1 lines 82-95	1, 4-6, 8-22
X	GB 2,022,378 A (TELEDIFFUSION) - see page 2 lines 45-77	1-6, 8-22
X	EP 0,511,438 A1 (IBM) - see column 3 lines 34-40	1-6, 8-22
X	MacWEEK, Vol 10, No 15, April 15, 1996, Rebecca Gulick, "Vendors vie for Web font standard", pages 12-13	1-22

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

PNSDOCID: <GB 2316778A >

This Page Blank (uspto)